

UNIVERSITATEA BABEȘ-BOLYAI
Facultatea de Științe Economice și Gestiunea Afacerilor
Informatică Economică

Lucrare de licență

Absolvent,
Gabriela-Diana **TĂTAR**

Coordonator științific,
Conf. univ. dr. Dan-Andrei **SITAR-TĂUT**

2023

UNIVERSITATEA BABEȘ-BOLYAI
Facultatea de Științe Economice și Gestiunea Afacerilor
Informatică Economică

Lucrare de licență

Digitalizarea fluxului de lucru dintr-o clinică de
oncologie în Flask cu baza de date MySQL

Absolvent,
Gabriela-Diana **TĂTAR**

Coordonator științific,
Conf. univ. dr. Dan-Andrei **SITAR-TĂUT**

2023

Rezumat

În cadrul acestei lucrări de licență am abordat dezvoltarea unei clinici de oncologie. Aceasta are scopul de a facilita accesul la servicii medicale pentru pacienții cu afecțiuni oncologice. Prin intermediul platformei, persoanele cu probleme de sănătate pot vizualiza profilul cadrelor medicale (medici, nutriționiști, psihologi), putând astfel să își facă programare la cel mai potrivit pentru ei, unde vor primi sfaturi de la experți. De asemenea, au o fișă medicală online cu datele lor, dar și un istoric medical pentru a putea ține evidența evoluției afecțiunii. În plus, personalul medical are acces la consultații, dar și disponibilitate la o listă cu pacienți pentru o cât mai bună contorizare a lor. Cu această aplicație web, sperăm să îmbunătățim serviciile de asistență medicală, oferind soluții actuale și convenabile atât pacienților, cât și personalului medical. Pe viitor ne dorim să implementăm mai multe funcționalități, cum ar fi un chat care să ofere pacienților răspunsuri la întrebările lor. Aplicația web este alcătuită din baza de date MySQL, back-end-ul realizat în limbajul Python, framework-ul Flask, iar front-end-ul este realizat cu limbajului Angular.

Cuprins

Abrevieri.....	iv
Glosar	v
Lista tabelelor și figurilor	vi
Introducere.....	1
1. Analiza sistemului informatic	4
1.1 Identificarea și descrierea problemei	4
1.2 Motivație	5
1.3 Obiective	8
1.4 Context	11
1.4.1 Fațeta Subiect	12
1.4.2 Fațeta Utilizare	12
1.4.3 Fațeta IT	13
1.4.4 Fațeta Dezvoltare.....	14
2. Cerințe de sistem	15
2.1 Surse de cerință.....	15
2.2 Elicitația cerințelor.....	15
2.2.1 Metoda interviului	16
2.2.2 Metoda Brainstorming	19
2.2.3 Metoda Cazurilor de utilizare (Use Case)	20
2.3 Formularea și documentarea cerințelor.....	25
2.3.1 Documentarea cerințelor.....	25
2.3.2 Procese și activități	26
2.3.3 Stările și tranzițiile sistemului.....	26
3. Proiectarea sistemului informatic.....	28
3.1 Proiectarea logică.....	28
3.1.1 Arhitectura sistemului	31
3.1.2 Baza informațională	34
3.2 Proiectare tehnică.....	34
3.2.1 Structura fizică a datelor.....	34
3.2.2 Procese și algoritmi	37
3.2.3 Tehnologii specifice	38
4. Implementarea sistemului informatic.....	41
4.1 Mediul de programare	41
4.2.1 Pregătirea bazei de date	41
4.2.2 Pregătirea zonei de back-end	43
4.2.3 Pregătirea zonei de front-end	48
5. Testarea platformei	55
5.1 Date generale legate de testare	55
5.1.1 Testarea de utilizabilitate	55
5.1.2 Testarea de compatibilitate	55
5.1.3 Testarea de integritate a sistemului	56
5.1.4 Testarea de funcționalitate	56
5.2 Cazuri de testare	56
Concluzii.....	61
Anexe	63
Bibliografie	64

Abrevieri

<i>API,</i>	Application Programming Interface
<i>CMD,</i>	linia de comandă
<i>CSS,</i>	Cascading Style Sheets
<i>DFD,</i>	Diagrama de Flux de Date
<i>EUPHA,</i>	European Public Health Association
<i>IOM,</i>	Institute of Medicine
<i>IT,</i>	Tehnologia informației
<i>HIT,</i>	Tehnologii Informaționale Medicale
<i>HTML,</i>	HyperText Markup Language
<i>HTTP,</i>	HyperText Transfer Protocol
<i>IDE,</i>	Integrated Development Environment
<i>TIC,</i>	Tehnologia informației și a comunicării
<i>WSGI,</i>	Web Server Gateway Interface

Glosar

API - Un set de reguli și protocoale care permit aplicațiilor să comunice și să interacționeze între ele, facilitând schimbul de informații și funcționalități.

Baza de date relațională - Un tip de bază de date care stochează și permite accesul la date legate între ele.

Business - Activitatea desfășurată în cadrul unei companii sau organizații în scopul generării de profit și satisfacerea nevoilor clienților.

Dezvoltatorul - Persoana responsabilă de crearea și menținerea unui site web.

IDE - Un mediu de dezvoltare integrat care oferă un set de instrumente și funcționalități pentru programatori, pentru a scrie, testa și depana codul software.

Interfața - Un punct de interacțiune între diferite componente sau dispozitive ale unui sistem, care facilitează comunicația bidirecțională prin intermediul compatibilității hardware și software.

Mock data - Date fictive sau simulate utilizate în timpul dezvoltării și testării unei aplicații pentru a înlocui datele reale.

Token/ Jeton de autentificare - O unitate de identificare sau acces care poate fi utilizată pentru a autentifica sau autoriza un utilizator într-un sistem sau aplicație.

Stakeholderul - Un individ sau o grupare cu un interes semnificativ în activitățile și rezultatele unei companii.

Lista tabelelor și figurilor

Tabele:

Tabel 1. Actorii cazurilor de utilizare	21
Tabel 2. Descrierea cazurilor de utilizare.....	23
Tabel 3. Test Case 1 - Testarea butonului de autentificare	57
Tabel 4. Test Case 2 - Testarea butonului de programare.....	57
Tabel 5. Test Case 3 - Testarea butonului de programare.....	58

Figuri:

Figura 1. Diagrama Fishbone	7
Figura 2. Diagrama de descompunere a obiectivelor (Archimate)	9
Figura 3. Business Model Canvas.....	11
Figura 4. Grafic Pareto	19
Figura 5. Cartografiere mentală	20
Figura 6. Diagrama cazurilor de utilizare (Use-Case)	24
Figura 7. Diagrama de activități din perspectiva pacientului	26
Figura 8. Diagrama de stări	27
Figura 9. Diagrama flux de date (DFD)	29
Figura 10. Diagrama entităților și relațiilor.....	30
Figura 11. Model arhitectură de sistem pe 3 niveluri	31
Figura 12. Diagrama pe trei niveluri detaliată	32
Figura 13. Diagrama de componente	33
Figura 14. Diagrama deployment	33
Figura 15. Diagrama de date.....	34
Figura 16. Structura tabelelor din designer-ul din phpMyAdmin	36
Figura 17. Modelul de înregistrare explicat	37
Figura 18. Modelul de autentificare explicat.....	38
Figura 19. Conectare la baza de date	42
Figura 20. Vizualizare tabele din phpMyAdmin	43
Figura 21. Vizualizare inițializare Flask	43
Figura 22. Vizualizare creare tabel Utilizatori	44
Figura 23. Creare metodă POST, pentru tabela Nutriționiști	45
Figura 24. Creare metodă GET, pentru tabela Planuri alimentare.....	46
Figura 25. Creare metodă GET/id, pentru tabela Planuri alimentare.....	46
Figura 26. Creare metodă DELETE, pentru tabela Utilizatori/public_id	47
Figura 27. Creare metodă PUT, pentru tabela Consultații/id_consultație.....	47
Figura 28. Definire jeton de autentificare (token)	48
Figura 29. Pagină inițială Angular	49
Figura 30. Rutele paginilor web	50
Figura 31. Vizualizare rute GET și POST.....	51
Figura 32. Pagina Acasă - rol vizitator.....	51
Figura 33. Pagina Acasă - rol cadru medical.....	52
Figura 34. Pagina Acasă - rol pacient	52
Figura 35. Pagina Consultatii	53
Figura 36. Pagina Planuri alimentare	53
Figura 37. Pagina Programările mele - pacient	54

Figura 38. Testare rută POST	58
Figura 39. Testare rută GET	59
Figura 40. Testare rută GET/id	59
Figura 41. Testare rută PUT	60

Introducere

În urma unor căutări mai amănunțite, am observat că în România sunt puține clinici de oncologie, care să ofere servicii în toate domeniile necesare, cum ar fi Sanador și Medisprof. Pacienții oncologici trebuie să fie monitorizați atât de către un medic, cât și de un psiholog care să îl ajute să gestioneze cât mai optim situația în care se află, dar și de către un nutriționist pentru a le oferi sfaturi și a le da un regim alimentar pentru o dietă adecvată stilului de viață. Din nefericire, pacientul trebuie să se plimbe la fiecare medic în parte, să își facă programare separat, dar cea mai important aspect care ar trebui schimbat este faptul ca prescripțiile sunt furnizate sub formă de document scris, pe o foaie de hârtie. Astfel, am ales să dezvolt o aplicație web pentru o clinică de oncologie, deoarece este important și relevant îmbunătățirea substanțială a calităților serviciilor medicale în acest domeniu. Prin intermediul platformei, am încercat să facilităm comunicarea dintre pacienți și cadrele medicale, reducând astfel durata de așteptare și timpul petrecut pentru rezolvarea problemelor administrative. Aceste aplicații au scopul de a ajuta persoanele cu afecțiuni oncologice să aibă acces la informații relevante despre clinică, despre serviciile oferite, dar și echipa medicală, ceea ce poate îmbunătăți gradul de încredere și satisfacție al persoanelor aflate în cauză.

Programul are acces la un sistem de programări online. Pacienții pot vedea disponibilitatea cadrelor medicale, vor găsi informații despre medicii din clinică, care sunt specializați în diferite categorii. De asemenea, se oferă date și despre nutriționiști și psihologi care au un rol important în viața unui pacient care suferă de afecțiuni oncologice. Pe de altă parte, serviciul online propus este unul benefic și pentru cadrele medicale, având o interfață centralizată pentru gestionarea programărilor. Una dintre cele mai importante calități ale aplicației este reprezentată de sistemizarea rapoartelor medicale ale pacienților pentru o organizare cât mai bună a echipei medicale. Așadar, consider că este util ca pe piața din România să existe cât mai multe clinici de oncologie care să cuprindă tot ce am menționat mai sus. Mi se pare o de o valoare esențială să încercăm să transformăm în format digital pentru o funcționare cât mai bună a lucrurilor. În plus, dezvoltarea unei astfel de aplicații web pentru o clinică de oncologie poate fi un pas cu efect vital către îmbunătățirea serviciilor medicale și creșterea satisfacției pacienților și a cadrelor medicale.

Cele trei aspecte care stau la baza îngrijirii moderne ale cancerului sunt următoarele:

- medicină clinică de performanțe ridicate, care pune în aplicare terapii avansate personalizate în funcție de caracteristicile specifice ale tumorii și ale pacienților;

- se urmăresc cu atenție nevoile pacienților, inclusiv cele fizice, psihosociale, funcționale și spirituale;
- prin intervenții umane cât și tehnologice se utilizează soluții sistematice, care ajută clinica să își atingă obiectivele și să ofere o îngrijire atentă asupra pacientului.

Pentru a optimiza caracteristicile menționate mai sus, se presupune ca deciziile să fie cât mai complexe, alături de o coordonare eficientă între cadrele medicale. S-au realizat mai multe rapoarte care descriu aspectul îngrijirii moderne, specializate pe cancer mai ales. Acestea au fost emise de IOM (Institute of Medicine), iar din păcate nu s-au focalizat pentru a demonstra cât de utilă poate fi implementarea tehnologiei informației (IT). (Clauser, 2011) Așadar, în urma documentării ne dorim să implementăm cele trei aspecte fundamentale, pentru a avea o clinică care pune accent pe nevoile pacienților, acela de a găsi mai multe informații centralizate într-o platformă.

În cadrul aplicației web pentru o clinică de oncologie, am utilizat mai multe tehnologii pentru a asigura funcționalitatea și performanța optimă. Pentru partea de back-end, am folosit limbajul de programare Python împreună cu framework-ul Flask. Python ne-a oferit flexibilitate și ușurință în dezvoltarea aplicației, iar Flask a facilitat gestionarea logicii și funcționalității. Acest framework este utilizat pentru configurarea sa simplă și capacitatea de a se extinde la aplicații mai complexe. În ceea ce privește partea de front-end, am ales platforma Angular, bazată pe limbajul de programare TypeScript. Angular ne-a ajutat să construim o interfață interactivă și prietenoasă pentru utilizatori, utilizând HTML, CSS și JavaScript. Această platformă include un număr mare de componente și biblioteci care acoperă o gamă largă de caracteristici. Pentru stocarea datelor, am utilizat o bază de date relațională MySQL. Alegerea MySQL s-a datorat performanței sale și compatibilității cu Python și Flask. Am utilizat, de asemenea, interfața phpMyAdmin pentru gestionarea și vizualizarea bazei de date.

Aceste tehnologii au fost selectate în funcție de avantajele și capacitățile lor în dezvoltarea unei platforme bazate pe o clinică de oncologie. Ele ne-au permis să implementăm funcționalități esențiale, cum ar fi gestionarea programărilor, comunicarea între pacienți și cadre medicale, și stocarea și accesul rapid la datele pacienților. În plus, unele dintre ele le-am învățat în cadrul facultății, de exemplu limbajul Python. Interfața phpMyAdmin am utilizat-o pentru diferite proiecte. Limbajul MySQL l-am asimilat la fel, în cadrul facultății, unde chiar am susținut cursurile oferite de platforma Oracle.

Această lucrare este organizată pe capitole și subcapitole, fiecare capitol și subcapitol având un nume sugestiv în raport cu materialele regăsite în el pentru a avea un ușor acces la

informația căutată. Mai mult, cuprinsul facilitează accesul la informația căutată și include elemente precum abrevieri, glosar, lista tabelelor și a figurilor. Lucrarea începe cu o introducere și continuă cu analiza sistemului informatic, cerințele de sistem, proiectarea sistemului, implementarea acestuia și testarea platformei. Lucrarea se încheie cu concluzii și direcții viitoare de implementat, iar materialele suport sunt afișate în cadrul Anexelor. De asemenea, bibliografia este inclusă la finalul lucrării.

1. Analiza sistemului informatic

1.1 Identificarea și descrierea problemei

În ultima vreme, mai multe activități economice au avut de îndurat schimbări majore, din cauza creșterii rapide a progresului în tehnologia informației și a comunicării (TIC). Tot mai multe companii, aleg să își facă publicitate printr-o platformă online.

Un grupul multidisciplinar de experți fondat de către Comisia Europeană a făcut o analiză referitoare la impactul tehnologiilor digitale asupra serviciilor de sănătate, pentru a identifica cele mai bune metode de a investi în infrastructura medicală. Aceștia nu au oferit o evaluare completă, ci doar o analiză a elementelor cheie ale digitalizării serviciilor de sănătate. Prin aceasta, ei recomandă modernizarea și sustenabilitatea tehnologiei medicale. (3rohealthreview, 2018)

O problemă complexă care va continua să aibă impact destul de mare asupra asistenței medicale este tehnologizarea acesteia, fiind un proces de convertire a datelor existente într-un format digital. Cu ajutorul digitalizării, va fi o organizare mai bună, influențând aspecte precum structurile de servicii, practicile profesionale, cultura asistenței medicale și rezultatele pacienților. Mai mult, se va îmbunătăți accesul la serviciile de sănătate, calitatea acestora, echitatea, rentabilitatea, sustenabilitatea și inovarea sistemelor de sănătate.

Noile tehnologii informaționale medicale (HIT) au adus o revoluție în sistemul de sănătate, deoarece se oferă diferite promisiuni cu privire la îmbunătățirea sănătății și a livrării serviciilor medicale. Ca exemplu este aplicația eHealth care crește accesul la informații utile, îmbunătățește calitatea asistenței medicale și încurajează comportamentele sănătoase. Totuși, proiectarea instrumentelor eHealth trebuie să ia în considerare nevoile utilizatorilor, dar și să fie ușor de folosit. (Kreps, 2010)

Datorită digitalizării sunt dezvoltate și aspecte de monitorizare continuă, tratamente și echipamente de îngrijire virtuală. Cu trecerea timpului, dar și cu implementarea a cât mai multor astfel de sisteme se va putea muta locul de tratament, în unele cazuri, chiar la pacient acasă.

1.2 Motivație

Prin aplicația realizată am atins un subiect sensibil, dar care trebuie abordat. De aceea, am ales ca motivația lucrării să fie servirea unui sistem medical de îngrijire a pacienților cu probleme oncologice. Persoanele care se confruntă cu această boală trebuie să li se acorde o atenție mult mai detaliată, iar cadrele medicale trebuie să se coordoneze adecvat deoarece trebuie să monitorizeze pacienții și să li se ofere un tratament cât mai potrivit nevoilor lor. Prin următoarele caracteristici, ne dorim să facilităm și să îmbunătățim procesul de îngrijire a unui pacient, oferindu-i toate resursele pentru o însănătoșire rapidă. Datorită unei sesiuni de brainstorming am reușit să colectez cât mai mulți factori care fac posibilă problema noastră, de îngrijire a pacienților. Prin intermediul următoarei figuri, Diagrama Fishbone, am găsit cauzele:

- Personal medical: aceasta se poate datora unui deficit de personal medical, lipsa de pregătire a acestora sau chiar nerespectarea protocoalelor. Din păcate, cele menționate mai sus sunt de multe ori niște probleme destul de mari în clinicile medicale, deoarece nu se mai găsește personal dornic de muncă, cu calificări demne pentru acest job.
- Echipamente și tehnologie: după un mic studiu realizat, am relatat ca este o problemă majoră a echipamentelor, pentru că se defectează frecvent, sau sunt insuficiente, sau chiar învechite. Mai mult, prin implementarea unui software care gestionează pacienții, s-ar reduce considerabil timpul de așteptare, dar mai ales ar fi o contorizare mai bună a datelor despre fiecare persoană.
- Factori externi: mai multe aspecte se iau în considerare când vine vorba despre acest subiect. Un factor extern este reprezentat de creșterea incidenței a persoanelor cu cancer. Pe de altă parte, și lipsa finanțării și a resurselor pentru îngrijirea pacienților cu probleme de oncologie este una importantă. Mai mult, putem discuta și din perspectivă politică, deoarece legile guvernamentale inadecvate și probleme legate de reglementare influențează major tema discutată.
- Procese și proceduri de lucru: de aici fac parte lipsa respectării protocoalelor de prevenire și control al infecțiilor, coordonarea inadecvată între departamentele medicale pentru a oferi un tratament cât mai optim, și în final, lipsa protocoalelor standardizate pentru îngrijirea pacienților cu cancer.
- Performanță: este datorată unor măsurători inexacte ale indicatoriilor medicali. În plus, performanța poate fi afectată și din cauza lipsei sistemelor de recenzie. În acest caz

pacientul nu are unde să trimită recenziile adecvate și meritate de către clinica respectivă, așadar nu se poate îmbunătăți pe urma experienței pacienților.

- Produse și servicii: este tratată superficial această cauză, pentru că persistă să apară în clinicile din România, aceasta fiind lipsa stocului de medicamente și consumabile necesare pentru îngrijirea persoanelor cu probleme oncologice. În al doilea rând, și produsele au de suferit în acest context, ca de exemplu lipsa disponibilității de materiale și servicii suport, cum ar fi servicii de nutriție și psihologie și lipsa accesului la tratamente noi și inovatoare.

DIGITALIZAREA FLUXULUI DE LUCRU DINTR-O CLINICĂ DE ONCOLOGIE

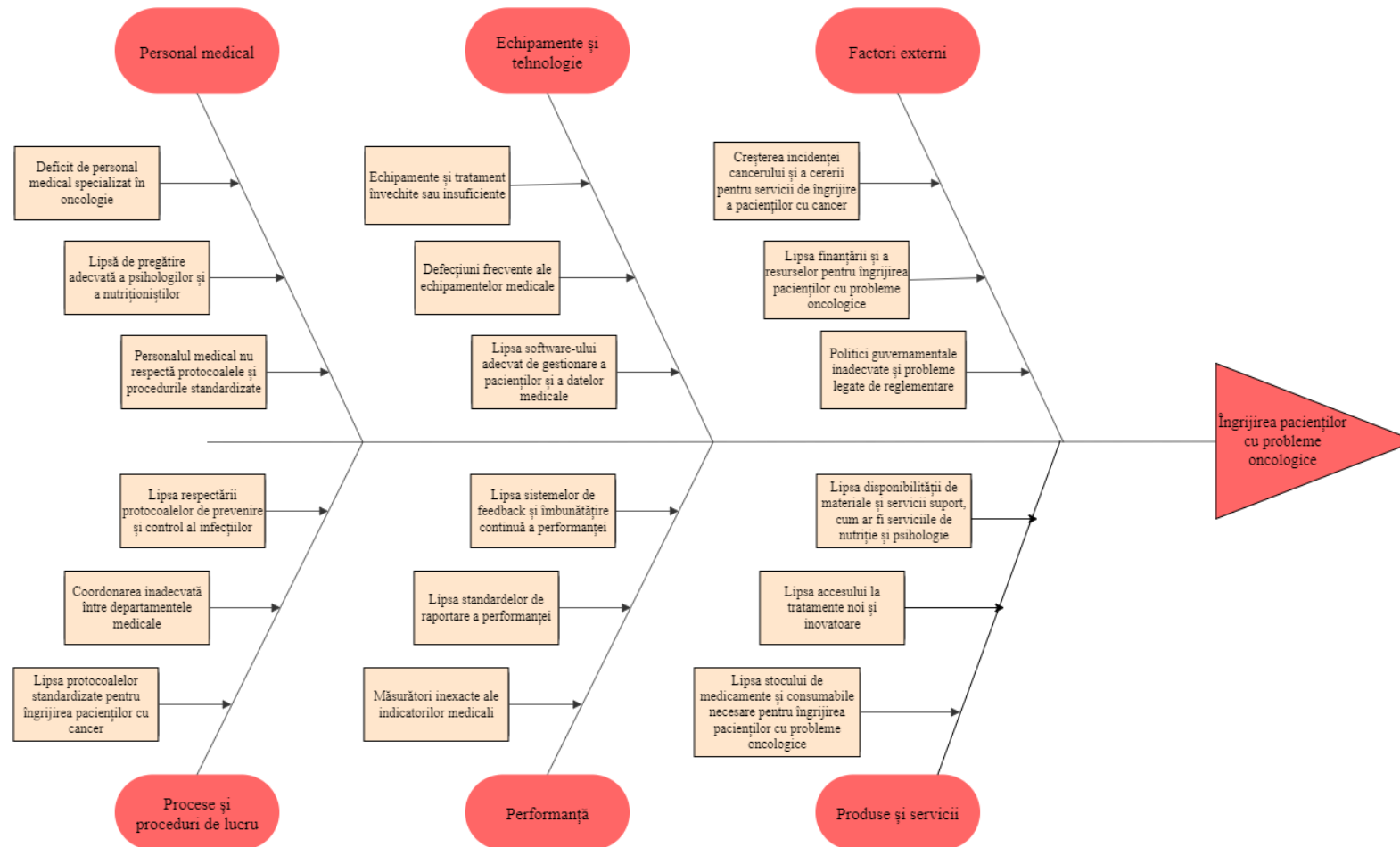


Figura 1. Diagrama Fishbone

1.3 Obiective

Pentru o cât mai bună acuratețe a aplicației, dar mai ales un succes al business-ului am implementat mai multe obiective. Ne-am ajutat de Figura 2. Diagrama de descompunere a obiectivelor (Archimate) pentru a le scoate în evidență.

- Din perspectiva pacienților

Cea mai importantă dorință pe care o avem este ca persoanele bolnave să se simtă confortabil și în singuranță în timpul tratamentului, dar și să oferim cele mai bune tratamente.

- Din punctul de vedere al cadrelor medicale

Medicii să poată dezvolta un program continuu pentru persoanele diagnosticate cu probleme oncologice.

Nutriționiștii să asigure servicii de nutriție adecvate, pentru contribuirea la eficacitatea tratamentului, dar și la menținerea unui stil de viață cât mai optim și sănătos.

Psihologicii să îi ajute pe pacienții, dar și pe familiile acestora cu servicii de consiliere psihologică.

Cu timpul, vrem să creștem numărul de cadre medicale, pentru a oferi cât mai multe servicii, dar mai ales să ne atingem scopul de a ajuta cât mai multă lume.

- Din punctul de vedere al mediului înconjurător

Deoarece aspectul ecologic a devenit foarte important, trebuie să învățăm să gestionăm deșeurile, să nu poluăm, și să folosim materiale cât mai prietenoase mediului.

- Din punctul de vedere al aplicației

Să putem dezvolta în timp aplicația cât mai mult, să ajungă în 2015 să aibă nota 5 în rating-ul Google. În plus, vrem să oferim o interfață cât mai intuitivă și ușor de utilizat pentru pacienți și personalul medical.

Prin implementarea aplicației, ne dorim să îmbunătățim satisfacția cadrelor medicale și a pacienților pentru a contura o relație cât mai pozitivă între acestea. Pentru a utiliza platforma va trebui doar să aibă acces la internet și la un browser. Datele medicale sunt stocate într-un

mediu cât mai securizat, pentru a nu întâmpina o neplăcere privind furtul de date sau pierderea lor.

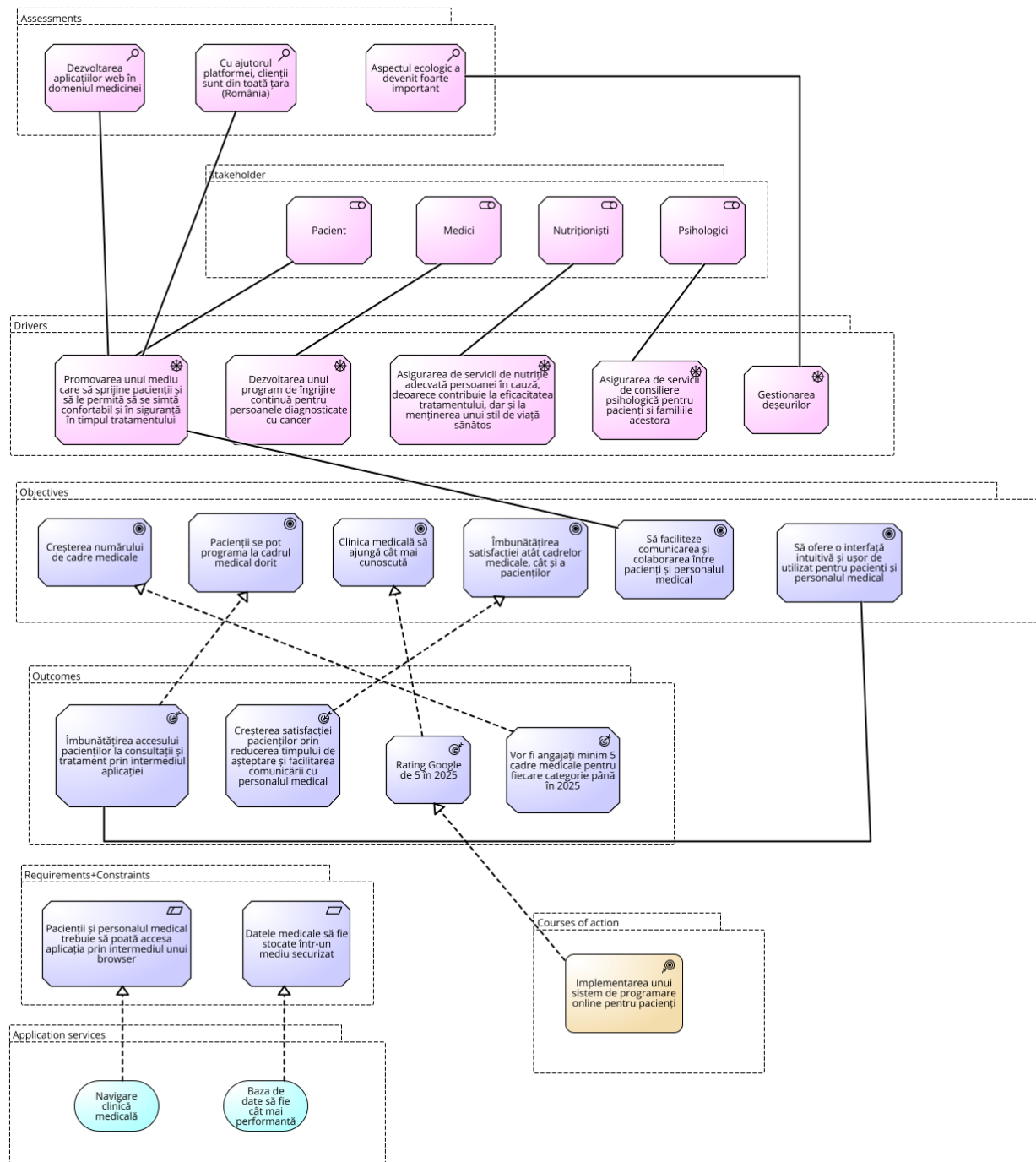


Figura 2. Diagrama de descompunere a obiectivelor (Archimate)

Prin folosirea diagramei Figura 3. Business Model Canvas, am reușit să descriem la nivel strategic începutul afacerii. Astfel ne-am folosit de cele 9 categorii:

- Partenerii cheie: am scos în evidență cadrele medicale cu experiență, care vor fi pline de viață și dornice de ajutor.
- Resursele cheie: ne-am legat de aplicația web care dispune de programări la doctori, psihologici și nutriționiști, având baza de date pentru stocarea și gestionarea informațiilor despre pacienți.
- Activitățile: aici se încadrează pașii pe care îi poți face în aplicație. Pacientul sau cadrul medical se poate înregistra în aplicație. Persoanele cu afecțiuni oncologice pot adăuga o programare în funcție de nevoile sale. Cadrele medicale vor realiza consultații medicale, servicii de nutriție și consiliere psihologica. Acestea după programare, vor întocmi sau vor actualiza rapoartele pacienților.
- Sursele de venit: se datorează plății tratamentului, a terapiei primită de la psiholog, dar și a planului alimentar primit de la nutriționist, de către pacient.
- Categoriile de cost: intervin situațiile salariale ale cadrelor medicale.
- Valoarea oferită pe piață: oferirea unei aplicații web care ajută persoanele cu probleme medicale oncologice să își găsească un cadru medical, dar și un tratament cât mai rapid. Mai mult, se va furniza un sistem de rapoarte eficiente pentru cadrele medicale pentru a înregistra progresul pacienților. Se va respecta cu strictețe măsurile impuse, deoarece ne dorim un mediu cât mai sănătos.
- Segmentele de piață țintite: pacienții cu afecțiuni oncologice care au nevoie de îngrijire medicală specializată, venită din partea medicilor, a nutriționiștilor, dar și a psihologilor.
- Relațiile cu clienții: cu ajutorul platformei dorim să creăm relații cât mai deschise și calde între pacienți și cadre medicale.
- Canalele de interacțiune: interacțiunea va fi una “față în față” în cadrul unui consult pe care îl programează din platforma web.

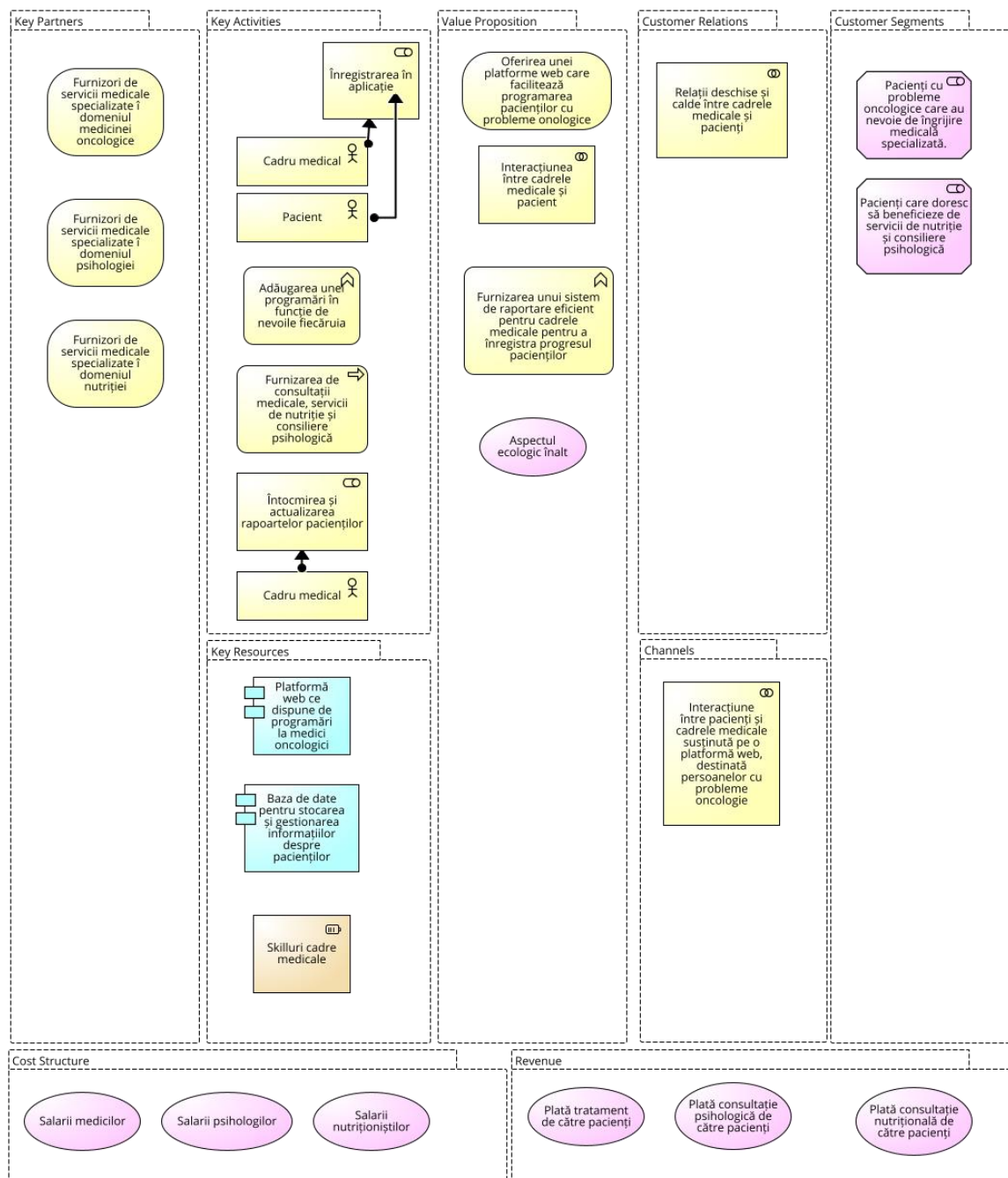


Figura 3. Business Model Canvas

1.4 Context

În acest capitol am descris contextul problematicei prezentate ulterior, folosind metodologia fațetelor. Cu ajutorul acesteia am prezentat soluția din patru abordări diferite. Sistemul nostru expune fațetele Subiect, Utilizare, IT și Dezvoltare.

1.4.1 Fațeta Subiect

Fațeta Subiect abordează scopurile și evenimentele din cadrul aplicației, având miza de a le trata. Din punctul de vedere al lucrării acestea sunt mai exact tratarea persoanelor cu afecțiuni oncologice.

Pe de o parte, un obiectiv pe care dorim să-l atingem este digitalizarea fluxului de lucru dintr-o clinică de oncologie. Fiind mult mai ușor și comod programarea la medic dintr-o aplicație, vizualizarea cadrelor medicale, dar și a detaliilor despre acestea, vrem să aducem totul într-o platformă web. Mai mult, din perspectiva medicilor este câteodată copleșitor a ține toată informația despre pacienții pe niște hârtii în dulap, dar și de modă veche. Așadar, dintr-o retrospectivă generală, ambele părți s-ar bucura de ajutorul digitalizării.

Pe de altă parte, un alt obiectiv al aplicației constă în oferirea celor mai bune tratamente pacienților. De aceea vor avea parte de îngrijire și din partea unor psihologici și nutriționiști. Cadrele medicale vor avea o pregătire pe măsură pentru a îndeplini cerințele clinicii. Cu trecerea timpului, ne dorim să mai angajăm persoane în domeniul, am putea spune să fie minim 5 angajați pentru fiecare categorie până în anul 2025.

Suplimentar un obiectiv pe care îl urmărește clinica noastră este acela de a pune accentul pe relația dintre cele două părți, să fie una cât mai caldă și plăcută, cu o atmosferă de învidiat. Este important să promovăm un mediu cât mai bun, pentru a sprijini pacienții și a le permite singuranța alegerii clinicii noastre, mai ales în timpul tratamentului. Se încercă facilitarea comunicării și a colaborării între cele două părți.

1.4.2 Fațeta Utilizare

Fațeta Utilizare face referire la aspectele legate de utilizarea aplicației de către utilizatori, pentru a atinge un set de obiective. În cadrul aplicației web avem trei tipuri de utilizatori, fiecare având diferite permisiuni și restricții. Cei trei participanți sunt: cadru medical, pacient și vizitator.

Cadru medical are un rol important în ceea ce privește domeniul oncologic, de aceea ne-am dorit să includem pe lângă medici care sunt prezenți la orice clinică, și nutriționiști și psihologici. Aceștia poate vedea o listă cu pacienți săi și informațiile acestora. În plus, după terminarea consultației fiecare va completa fișa medicală a persoanei în cauză cu toate detaliile

aferente. Pentru crearea unui cont a cadrului medical se va realiza direct din baza de date. Cel menționat va avea acces la următoarele pagini:

- Acasă
- Autentificare
- Acasă după autentificare (se afișează detaliile medicului)
- Listă pacienți
- Pagină detalii pacient
- Consultații

Pacientul își poate face cont în aplicația web destinată persoanelor cu afecțiuni oncologice, din pagina Înregistrare, urmând să se autentifice pe site din pagina de Autentificare. Poate accesa o listă cu categoriile medicilor, selectând una. Pe urmă va putea să se programeze la medicul dorit, completând niște câmpuri. La fel va fi și pentru nutriționiști sau psihologi. În plus, pacientul are o pagină cu consultațiile realizate, dar și una cu cele din viitor. Pentru a se putea informa cât mai bine, are accesibilitatea să citească lucruri despre Tratamente, Planuri alimentare și Terapii.

Vizitatorul are un acces destul de mic, asupra platformei, putând să vizualizeze detaliile despre clinică în pagina de acasă. Se poate documenta din perspectiva cadrelor medicale aflate în clinică, văzându-le caracteristicile acestora. O misiune importantă este aceea de a atrage un număr cât mai ridicat de vizitatori, în așa fel încât aceștia să apeleze la serviciile puse la dispoziție de medicii noștri.

1.4.3 Fațeta IT

Fațeta IT ilustrează platforma web Clinică de oncologie, compusă din următoarele limbaje:

- Bază de date s-a realizat în MySQL, cu ajutorul interfeței phpMyAdmin;
- Back-end-ul folosit Python, framework Flask;
- Front-end-ul utilizat Angular, cu toate componentele sale, iar pentru un design mai frumos ne-am ajutat de Bootstrap.

Pentru a beneficia de aplicație trebuie impuse câteva condiții, cum ar fi:

- Conexiune la internet;
- Pentru o funcționare optimă se recomandă o versiune a browser-ului cât mai recentă, indiferent care ar fi (de exemplu: Microsoft Edge, Chrome, Internet Explorer);
- Nu se impune o specificare a sistemului de operare, deoarece rulează pe toate.

Clinica medicală nu va distribui niciun detaliu legat de pacienți, pentru respectarea cu prevedere a normelor GDPR.

1.4.4 Fațeta Dezvoltare

Fațeta Dezvoltare înfățișează îmbunătățirile care pot fi aduse aplicației web, oricărui lucru i se poate aduce un progres. În viitor ne dorim să implementăm aplicația și pentru telefon, ca să ne extindem clinica cât mai mult. Un alt obiectiv pe care ne concentrăm este ca să creăm satisfacție pacienților, implementând o secțiune unde aceștia vor putea lăsa recenzii și evaluare. O metodă de implementare este un chat online, care să permită pacienților să comunice direct cu medicii sau personalul clinicii. Aceasta poate fi utilă pentru întrebări rapide sau pentru a solicita sfaturi înainte sau după vizitele medicale.

Cu trecerea timpului, dorim să facem o clinică cât mai optimă și complexă, de aceea vrem să implementăm și un laborator în cadrul clinicii. Așadar, se va dezvolta și în aplicația web o pagină pentru a te programa pentru a urma niște analize, și o altă pagină pentru primirea și vizualizarea lor.

2. Cerințe de sistem

2.1 Surse de cerință

În legătură cu aplicația web dezvoltată, sursele de cerințe au fost precis identificate, urmând să aibă un scop bine întemeiat. Acestea înfățișează menținerea pacienților fericiți și o colaborare cât mai bună între cadrele medicale și persoanele afectate de probleme oncologice. Datorită acestor obiective aplicația are un viitor cât mai grandios.

În primul rând, ne dorim ca și clinica să aibă cât mai mulți pacienți, pentru a produce un profit pe care în viitorul apropiat să îl investim în modernizarea aplicației, în calificarea cât mai profesională a medicilor, dar nu în ultimul rând în tratamente și echipamente destinate vindecării cancerului. În plus, vom cauta sponsori care doresc să investească în clinica medicală pentru a o îmbunătăți tot mai mult, pentru a fi cât mai optimă și eficientă.

În al doilea rând, cadrele medicale reprezintă un stakeholder interesat de aplicație, care doresc să își ofere cunoștințele pentru a salva lumea. Ele au accesibilitatea de a-și dezvolta îndeletnicirea și de a prinde experiență. Acești stakeholderi sunt medici, nutriționiști și psihologici. În cele din urmă, un alt stakeholder implicat este pacientul care dorește să primească cele mai bune investigații. Deoarece ne dorim să oferim cele mai de calitate servicii vom organiza ședințe lunare pentru a vedea ce am putea îmbunătăți cu toți stakeholderii.

2.2 Elicitația cerințelor

Părțile implicate în implementarea aplicației sunt utilizatorii care fac parte din două categorii: cadre medicale (medic, nutriționist, psiholog) și pacient. Ambele părți beneficiază în egală măsură de accesarea aplicației.

Metodele de elicitare pe care le am pus în aplicare sunt următoarele: metoda interviului, metoda brainstorming și metoda cazurilor de utilizare (Use Case).

2.2.1 Metoda interviului

Prima metodă pe care o prezentăm este Metoda interviului care are ca intenție obținerea de informații de la pacienți pentru a optimiza aplicația web. Rezultatele furnizate sunt informații fictive, cunoscute și sub denumirea de “mock data”, iar pe baza lor vom vedea satisfacția persoanelor care utilizează platforma. Această metodă este una puternică de a dezvălui deficiențe care pot fi ameliorate pe urmă. Mai jos vom exemplifica un model de interviu adresat unui pacient, bine înțeles cum am menționat și anterior datele nu sunt reale:

- Cum v-ați simțit în timpul utilizării aplicației web de clinică medicală?

Am simțit că aplicația era destul de ușor de folosit, butoanele fiind zărite rapid, ceea ce a crescut satisfacția. Nu am simțit frustrare sau nerăbdare. A fost o experiență plăcută și rapidă atât din punct de vedere al unui pacient, cât și al unui doctor.

- Ce aspecte ale aplicației considerați că au fost mai utile sau eficiente?

Cele mai utile aspecte au fost meniul din partea de sus a aplicației, fiind ușor de văzut și utilizat, categorizarea medicilor în funcție de tipul programării/serviciului oferit, categorizarea în pagini diferite ale programărilor în funcție de dată, sistemul ușor de creare programări deoarece este nevoie de foarte puține date pentru a cere o programare la clinică. Din punctul de vedere al unui medic, unele dintre cele mai eficiente/folositoare aspecte ale aplicației au fost sistemul ușor de adresare programări, fiind vorba de niște informații care se completează rapid în aplicație.

- Cum ați apreciat ușurința utilizării și navigarea în aplicație?

Meniul a ajutat foarte mult la navigare, majoritatea paginilor fiind prezente acolo, iar restul paginilor au fost ușori navigabile mulțumită butoanelor vizibile, puține la număr, care te duceau la următoarea destinație. În rest, aplicația era ușor folosibilă mulțumită modului în care a fost împărțită pe pagini, fiecare având un scop principal ușor discinbil.

- Ați întâmpinat dificultăți în înregistrarea sau programarea unui consult medical prin intermediul aplicației? Dacă da, care au fost acestea?

Nu am întâmpinat dificultăți, deoarece era vorba de puține informații de trecut pentru o nouă programare, doar data, medicul și simptomele. Pentru a afla ora te contactează personalul de

la clinică, iar aceasta ar putea fi o îmbunătățire pentru viitor, să poți alege și ora programării direct din aplicație.

- V-ați simțit în siguranță furnizând datele dvs. personale prin intermediul aplicației web?

Da, deoarece știu ce datele se află în posesia clinicii, într-o bază de date. Știu ca datele vor ajunge mai departe către cadrele medicale, motiv pentru care mă simt în siguranță să furnizez datele mele aplicației.

- Cum ați evaluat comunicarea și interacțiunea cu medicii sau personalul medical prin intermediul aplicației?

Sistemul de programări, istoricul de programări cu rețetele atașate reprezintă interacțiunea cu personalul clinicii prin intermediul aplicației. Evaluez aceste aspecte ca fiind ușor de folosit/citit, fiind destul pentru ce are nevoie un pacient.

- Ați găsit informațiile furnizate de aplicație privind planurile alimentare, tratamentele și terapiile aferente ca fiind utile și bine structurate?

Da, mi-a plăcut structura lor, informațiile fiind ușor de înțeles pentru un necunoscător de termeni medicali. Planurile alimentare erau însoțite de imagini, ceea ce ajută în citire și înțelegere. Faptul că erau adăugate imagini de câte ori erau explicate concepte pe înțelesul necunoscătorilor, este apreciat de către utilizatorii aplicației.

- Cum ați perceput nivelul de confidențialitate și securitate a datelor dvs. în cadrul aplicației?

Sunt sigură ca nivelul de confidențialitate era crescut pentru datele furnizate, majoritatea utilizatorilor având încredere să ofere informații personale fiind vorba de o clinică medicală care oferă un serviciu mult mai rapid online.

- Cum ați apreciat posibilitatea de a avea acces la istoricul medical și la informațiile dumenavoastră personale în aplicație?

Aceste lucruri au fost foarte apreciate în interacțiunea cu aplicația. Aplicația se deosebește de restul aplicațiilor medicale prin faptul ca îmbină psihologi, nutriționiști, medici oncologi, tot ce trebuie unei clinici de oncologie. Iar pe lângă acest lucru, ai acces la cele menționate mai sus, fapt care oferă o unicitate aplicației.

- Ce alte caracteristici sau funcționalități ați dori să fie adăugate sau îmbunătățite în aplicația de clinică medicală?

Un sistem de consult online ar fi bun pentru programările care nu ar necesita neapărat prezența fizică. Poate și o funcționalitate de spitalizări ar fi folositoare.

Ați recomanda aplicația de clinică medicală altor persoane sau prieteni?

Da, chiar dacă nu este un subiect la care mi-aș dori să le trebuiască acces. Aplicația este foarte folositoare și ușor de folosit.

După realizarea interviului, am urmărit problemele care ar putea apărea și am chestionat 100 de persoane care ne-au dat un răspuns bazat pe experiența lor. Cum am prezentat și mai sus datele sunt fictive. Am decizi să punem datele într-un grafic Pareto pentru a putea identifica cât mai clar cauzele. Ca aplicația să fie cât mai eficientă, a trebuit să investigăm principalele probleme și să identificăm cauzele acestora, ca în viitor să le ameliorăm. În urma interviului am realizat următoarele statistici:

- 40% dintre oameni au spus că motivul principal a fost găsirea unor clinici concurente mai apropiate de orașul lor
- 30% consider că este un număr mic de cadre medicale
- 15% au afirmat că principalul motiv îl reprezintă prețurile mici oferite de alte clinici
- 10% relatează că e o comunicare slabă între cadrele medicale (medic, nutritionist, psiholog) care necesită timp, și de aceea primesc greu prescripțiile
- 5% declară că motivele au fost altele decât cele prezentate mai sus

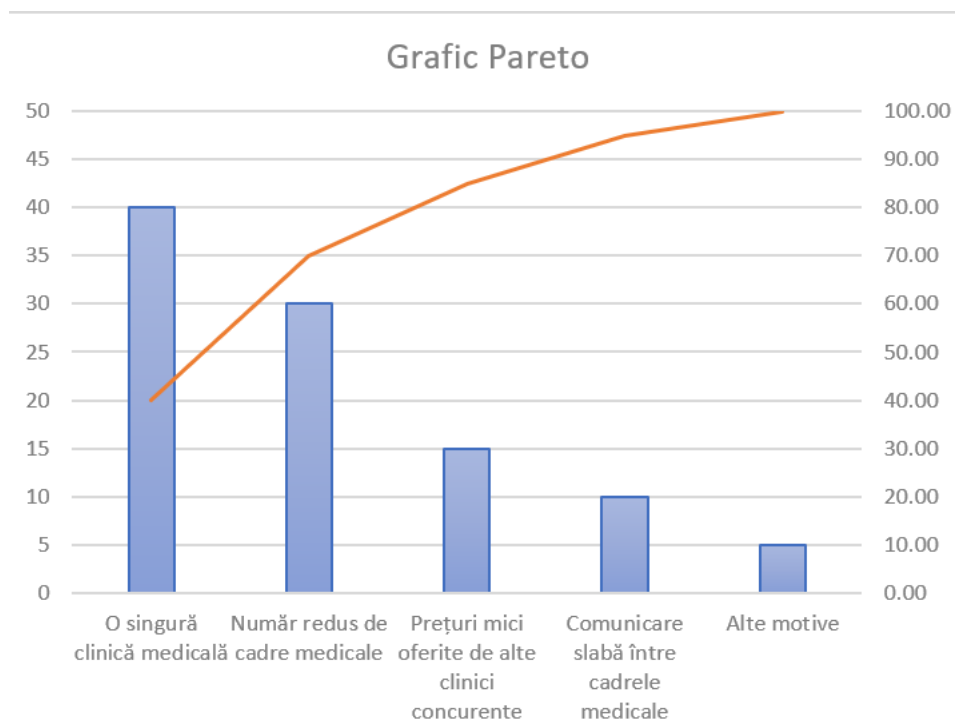


Figura 4. Grafic Pareto

2.2.2 Metoda Brainstorming

Această metodă de elicitare surprinde generarea unei game diverse de idei posibile pentru aplicația de clinică medicală de oncologie, fiind un proces creativ. Scopul sesiunii de brainstorming a fost stabilit încă de la început. Acesta ne-a permis să rezolvăm mai precis problemele specifice, precum îngrijirea persoanelor cu afecțiuni medicale.

Pentru a obține o aplicație inedită, am realizat o ședință formată din șase persoane: eu (dezvoltator aplicație), sora mea (angajată în cadrul unei companii de IT), Cristina (cunoștințe medicale), Andreea (cunoștințe nutriționale), Doriană (cunoștințe psihologie) și Irina (cunoștințe design). Am menționat membrii cheie ai echipei de dezvoltare și datorită întâlnirii ne-am permis să obținem o imagine de ansamblu completă și să aducem expertiza necesară pentru a dezvolta funcții relevante pentru fiecare categorie de utilizatori.

La începutul sesiunii de brainstorming am încurajat fiecare participant să gândească cât mai deschis și să își exprime liber ideile. Am folosit tehnica cartografiere mentală (“mind mapping”), deoarece ne ajută să organizăm propunerile într-un mod structurat și creativ. Se pornește de la un element central, plasat în centrul hărții și reprezintă focalizarea asupra

gândului principal. Apoi se construiesc diferite ramuri pentru a ne ajuta să construim un adevărat concept referitor la aplicația dezvoltată.

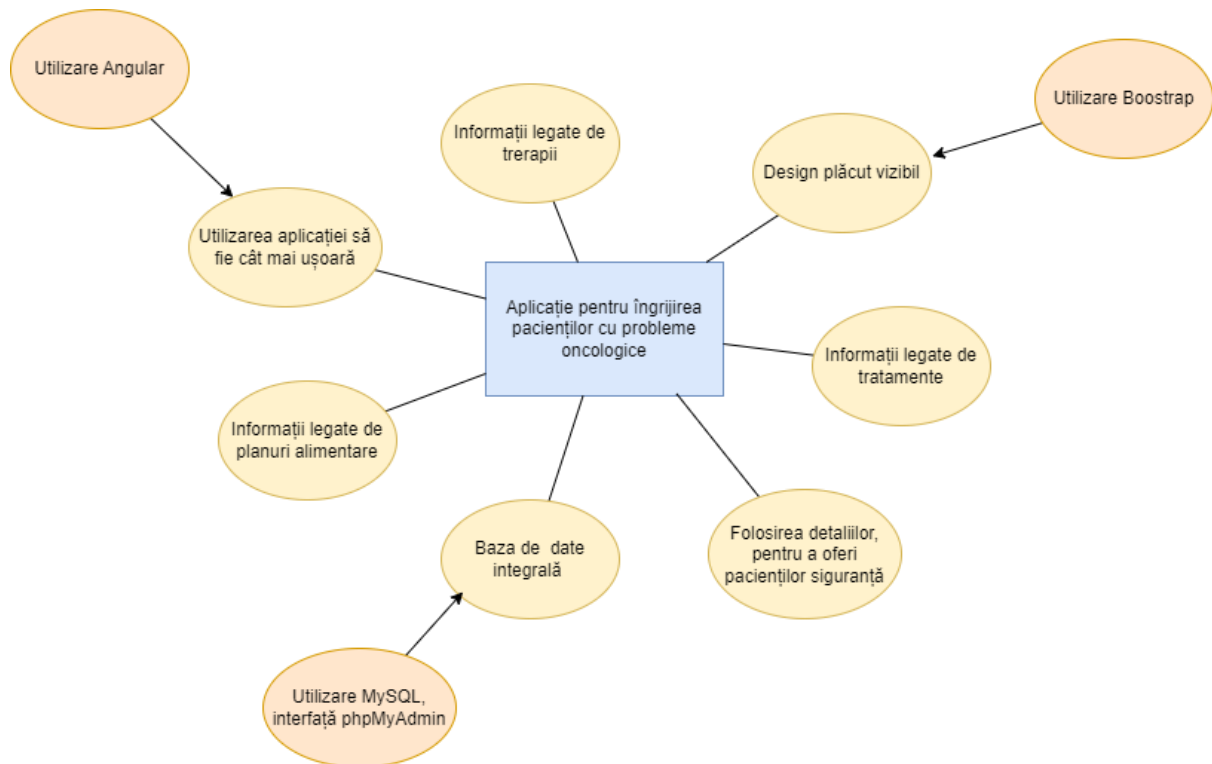


Figura 5. Cartografiere mentală

Am impus câteva reguli pentru o funcționare cât mai bună și o colaborare de succes între membrii echipei. Una dintre acestea ar fi dezvoltarea unui mediu deschis și fără judecăți. Altă regulă face referire la sprinjinul acordat unul celuilalt, primind încurajări reciproce în timpul procesului de brainstorming și nu numai.

La finalul ședinței, am început evaluare și selectarea ideilor de implementare în aplicație. Aceasta se bazează pe aspecte precum: relevanță pentru nevoile oamenilor, atingerea obiectivelor proiectului și potențialul de a îmbunătăți semnificativ experiența utilizatorului.

2.2.3 Metoda Cazurilor de utilizare (Use Case)

Misiunea acestei metode este de a determina ce tip de utilizatori folosesc platforma web și ce părți ale programului pot fi accesate de fiecare actor în parte. Aceștia reprezintă oamenii care interacționează cu aplicația, având două tipuri în proiect: utilizatori și vizitatori. Pentru a asigura funcționarea organizată a site-ului, fiecare actor are acces la anumite părți ale aplicației.

Utilizatorul simbolizează un actor care va beneficia de întregirea site-ului. Acesta nu va avea restricții de utilizare. Mai mult, utilizatorii se împart în două categorii: pacienți și cadre medicale. Pentru a se înregistra pe platforma web, pacientul își poate face cont chiar din interfață. Pe când, cadrelor medicale li se face cont din cadrul bazei de date de către dezvoltator, mai exact medicii, nutriționiștii și psihologii nu își vor face cont din aplicație.

Vizitatorul este restricționat la mai multe pagini. Acesta poate să acceseze lista cadrelor medicale, informații despre clinică, dar și tratamentele, planurile alimentare și terapiile utilizate de către clinică. Acesta se poate hotărî dacă vrea să își facă cont la clinica noastră medical prin intermediul conținutului afișat. Datorită acestui motiv, paginile menționate sunt create profesionist, deoarece vizitatorii pot ajunge oricând pacienții noștri.

Rol	Descriere	Obiective
Utilizator	Utilizatorul dispune de acces la întregul conținut al aplicației.	Folosește aplicația cu scopul de a avea o colaborare cât mai plăcută.
Vizitator	Vizitatorul dispune de acces la pagina publică de “Acasă”, dar și la înregistrarea sa pe site.	Vizualizează datele de pe prima pagina unde sunt informații despre clinică, iar cu această ocazie pot să se hotărască dacă vor să își facă cont sau nu.

Tabel 1. Actorii cazurilor de utilizare

Număr	Caz de utilizare	Descriere
1.	Vizualizare detalii clinică	Ca vizitator poți să vezi detalii legate de clinica medical de oncologie.
2.	Vizualizare cadre medicale	Ca vizitator poți să vezi detalii legate de cadrele medicale, pentru ați putea forma o idee despre serviciile furnizate.

Număr	Caz de utilizare	Descriere
3.	Vizualizare informații	Ca vizitator poți să vezi detalii legate de tratamentele, planurile alimentare și terapiile pe care clinica noastră le oferă.
4.	Înregistrare	Fiecare utilizator nou va trebui să completeze datele aferente pentru crearea unui cont nou. Mai exact: nume, prenume, CNP, numărul de telefon, email, județ, localitate, adresă, nume utilizator și parolă.
5.	Autentificare	După ce procesul de “Înregistrare” a fost finalizat, pacientul este nevoit să accese pagina de “Autentificare” pentru a avea acces în aplicație. Va trebui să introducă numele de utilizator și parola, deoarece logarea se face pe baza credențialelor create în procesul de înregistrare.
6.	Vizualizare consultații	Din perspectiva cadrelor medicale, acestea vor putea vedea consultațiile programate.
7.	Modificare raport medical	Din această pagină se va putea face modificarea prescripției după fiecare programare. Acces au doar cadrele medicale.
8.	Vizualizare listă pacienți	Se va afișa o listă cu toți pacienții care aparțin unui cadru medical.
9.	Pagină detalii pacient	Din lista, se va putea alege un pacient pentru a îi vedea detaliile sale.
10.	Vizualizare medici	Pacientul are acces la o listă cu medicii.
11.	Programare medic	Acesta se poate programa la medical dorit.
12.	Vizualizare nutriționiști	Pacientul are acces la o listă cu nutriționiști.

Număr	Caz de utilizare	Descriere
13.	Programare nutriționist	Acesta se poate programa la nutriționistul dorit.
14.	Vizualizare psihologi	Pacientul are acces la o listă cu psihologi.
15.	Programare psiholog	Acesta se poate programa la psihologul dorit.
16.	Vizualizare programările mele	În această pagină, pacientul poate vizualiza consultațiile pe care le are de onorat.
17.	Vizualizare istoric medical	Pacientul are acces la programările trecute, conform acestei pagini.
18.	Vizualizare planuri alimentare	Se afișează planurile alimentare pe care clinica medical le folosește.
19.	Vizualizare terapii	Prezintă terapiile folosite de psihologi.
20.	Vizualizare tratamente	Expune o gamă de tratamente, pentru vindecarea cancerului.

Tabel 2. Descrierea cazurilor de utilizare

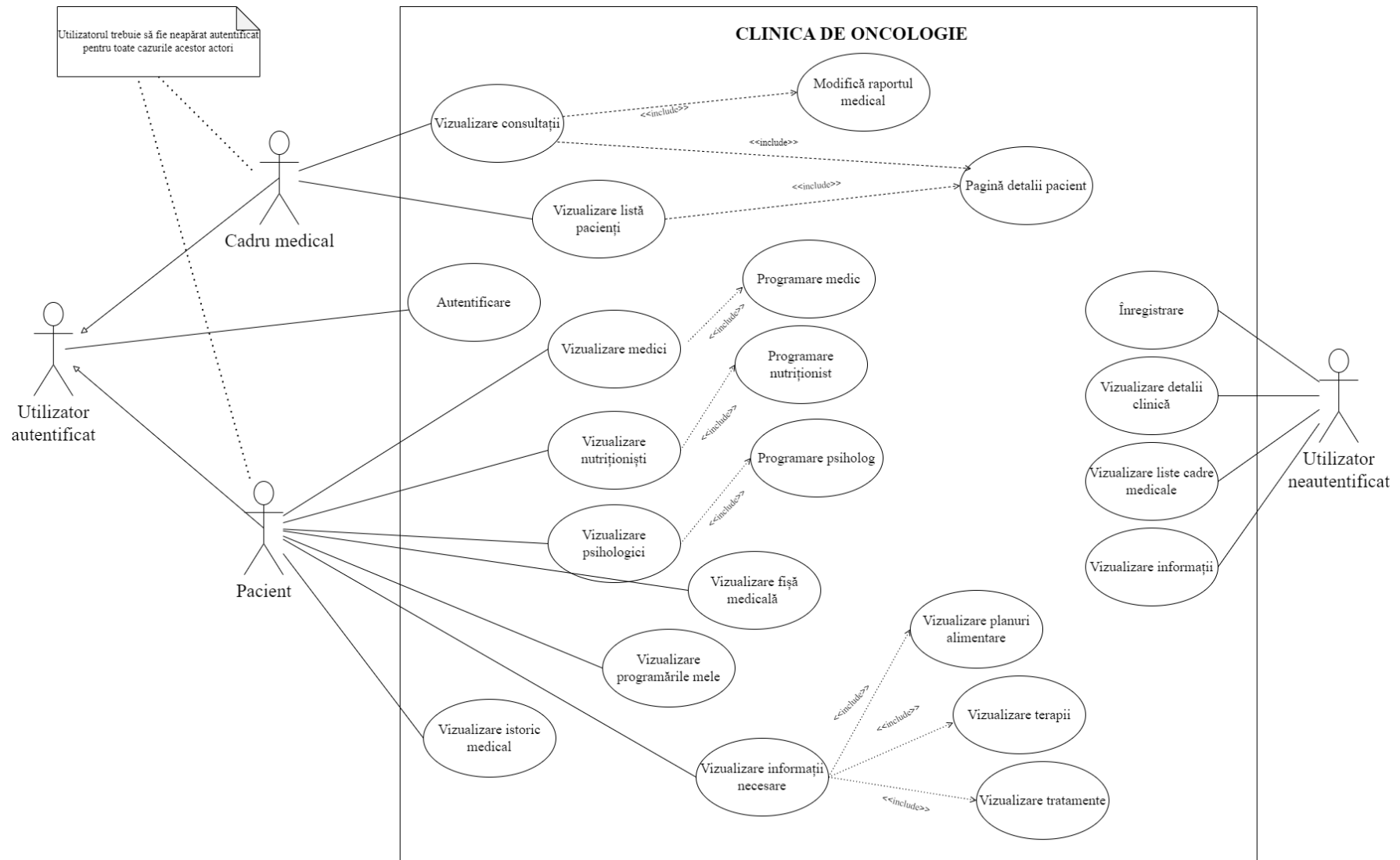


Figura 6. Diagrama cazurilor de utilizare (Use-Case)

2.3 Formularea și documentarea cerințelor

2.3.1 Documentarea cerințelor

Este important să abordăm funcția pe care trebuie să o îndeplinească programul. Am împărțit cerințele sistemului informatic în mai multe categorii pentru o prezentare generală, după cum urmează:

Cerințe de sistem

- Conexiune la internet este un factor principal.
- Acces la o variantă recentă de browser, indiferent care este. De exemplu Microsoft Edge, Chrome, Internet Explorer, Opera, Mozilla Firefox.
- Platforma trebuie să se comporte cât mai bine, așadar trebuie rezolvate toate erorile.

Cerințe funcționale

- Crearea unui cont de pacient se realizează dintr-un formular găsit în pagina de înregistrare.
- Cele două procese de logare și delogare se efectuează într-o manieră rapidă.
- Accesul la paginile publice se pot realiza din perspectiva vizitatorului.

Cerințe non-funcționale

- Tehnologia folosită pentru dezvoltarea aplicației este Visual Studio.
- Codul din spatele site-ului web este accesibil doar dezvoltatorilor, astfel încât niciun actor de orice tip nu îl poate accesa și edita.
- Datele sunt stocate în așa fel încât să nu poată fi pierdute, distruse sau modificate fără voința dezvoltatorului.

Cerințe calitative

- Interfață este plăcută atât din punct de vedere vizual, cât și din punct de vedere funcțional, fiind ușoară de manevrat.
- Se oferă eficiență și eficacitate.

2.3.2 Procese și activități

În acest subcapitol, prezentăm funcționarea aplicației web din perspectiva unui pacient. Aici se explică toate paginile pe care utilizatorul le poate parcurge, de la vizualizarea cadrelor medicale, la programarea unui consult.

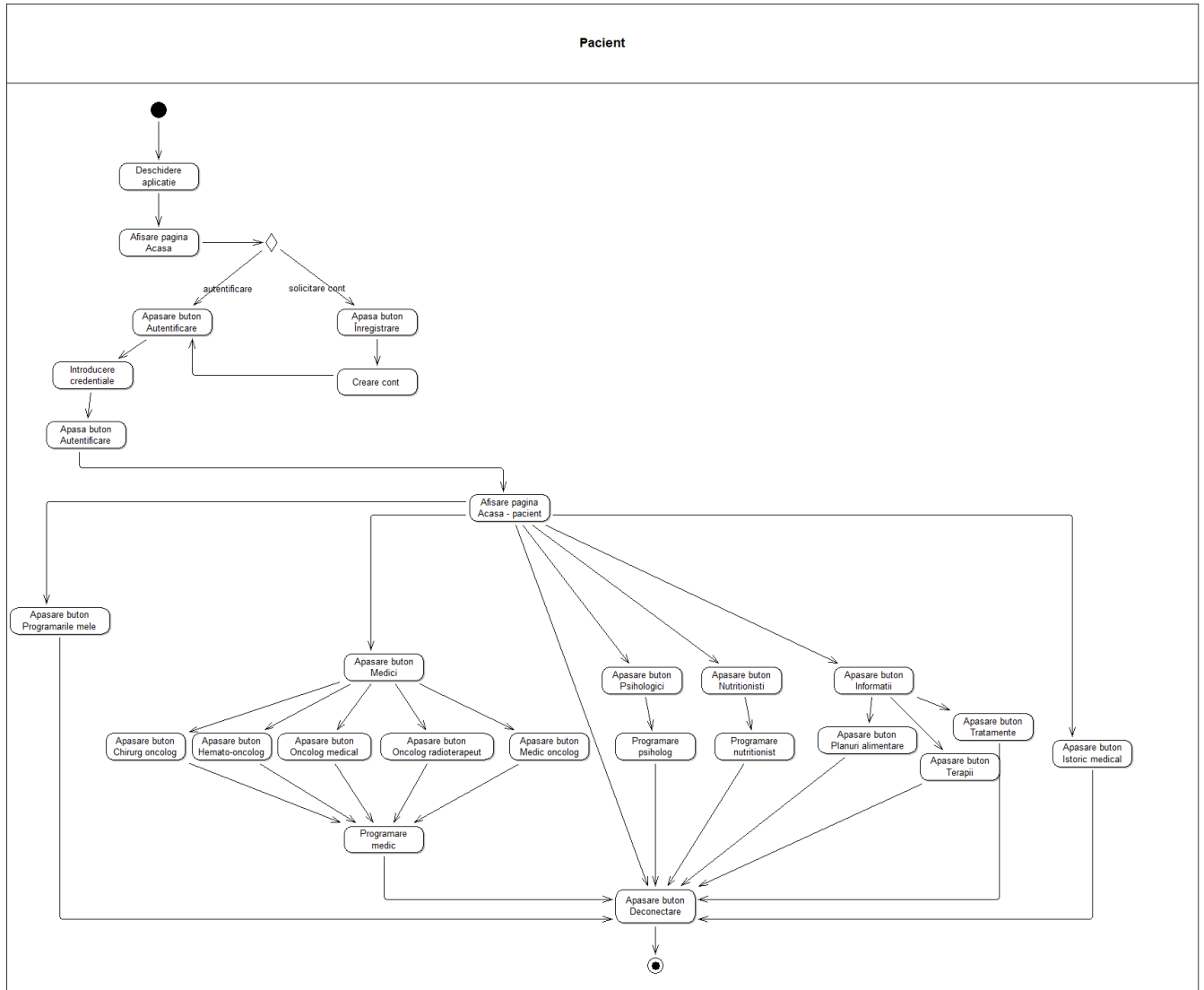


Figura 7. Diagrama de activități din perspectiva pacientului

2.3.3 Stările și tranzițiile sistemului

Mai jos exemplificăm Diagrama de stări care ne dezvăluie stările și tranzițiile sistemului. Diagrama începe cu o stare inițială și se finalizează cu o singură stare finală, iar pe parcurs între cele două se efectuează diferite tranziții.

DIGITALIZAREA FLUXULUI DE LUCRU DINTR-O CLINICĂ DE ONCOLOGIE

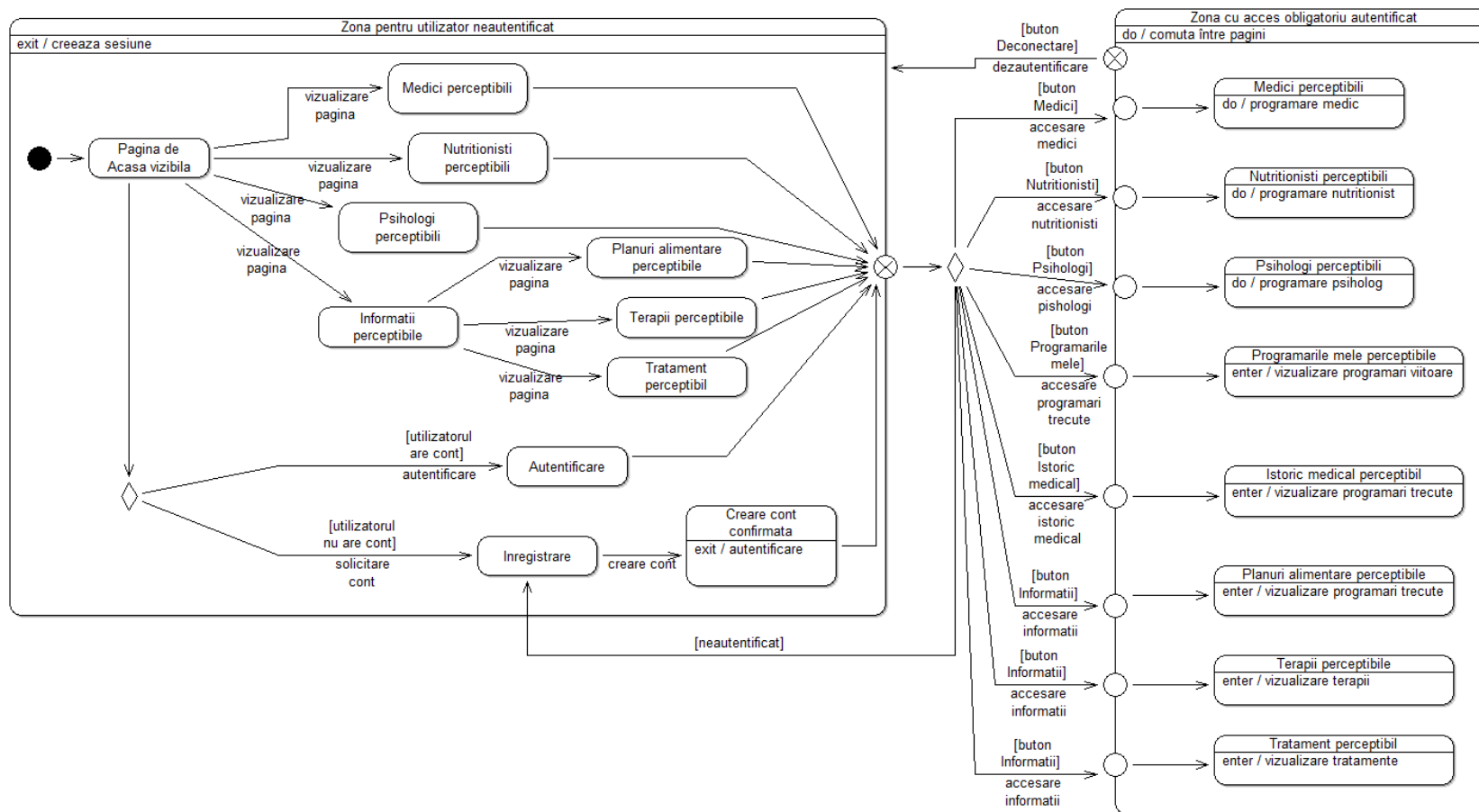


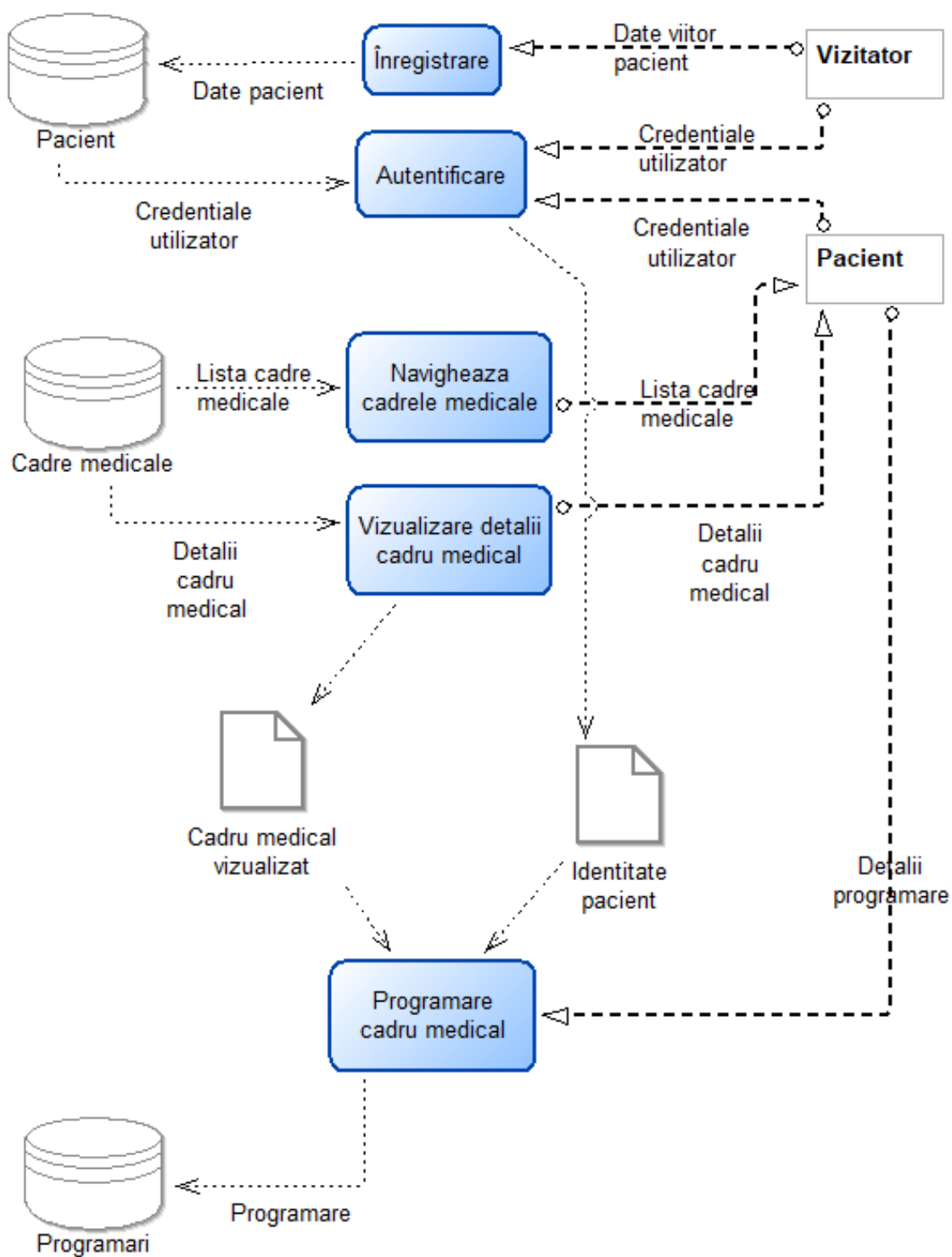
Figura 8. Diagrama de stări

3. Proiectarea sistemului informatic

3.1 Proiectarea logică

Aplicația web are în spate o singură bază de date care este amplu realizată conținând toate informațiile necesare. Am hotărât să implementăm în acest fel, deoarece putem astfel să reducem redundanța datelor și de a asigura că datele sunt stocate bine și doar o singură dată. În plus, în urma abordării respective poate apărea inconsecvența datelor, adică nu vor exista multiple copii ale informației stocate. Datorită acestui aspect îi este mai ușor și dezvoltatorului aplicației să aibă datele în același loc, ceea ce face să fie un acces mai rapid și ușor. Singura persoană care poate face o schimbare în baza de date este dezvoltatorul, deoarece el este îndrumătorul lucrării. Pacienții vor putea să își facă cont direct din platformă, iar cadrelor medicale le va face cont dezvoltatorul.

În continuare este scoasă în față împărțirea pe unități logice-funcționale a sistemului informatic. Este un concept care ajută la atingerea scopurilor, de exemplu cele impuse de cadrele medicale care doresc să ajute lumea și să își ofere toate cunoștințele și experiența pentru a face bine. Pentru a evidenția modul în care sunt structurate datele, mai exact proiectarea logică a lor, am utilizat Figura 9. Diagrama flux de date (DFD, efectuată în limbajul BPMN. Mai mult, vom arăta cu ajutorul diagramei cum sunt transferate datele, dar cum circulă acestea, pentru realizarea unui cont nou din punctul de vedere al unui vizitator, viitor pacient, și pentru programarea la un cadru medical dorit.

**Figura 9.** Diagrama flux de date (DFD)

DIGITALIZAREA FLUXULUI DE LUCRU DINTR-O CLINICĂ DE ONCOLOGIE

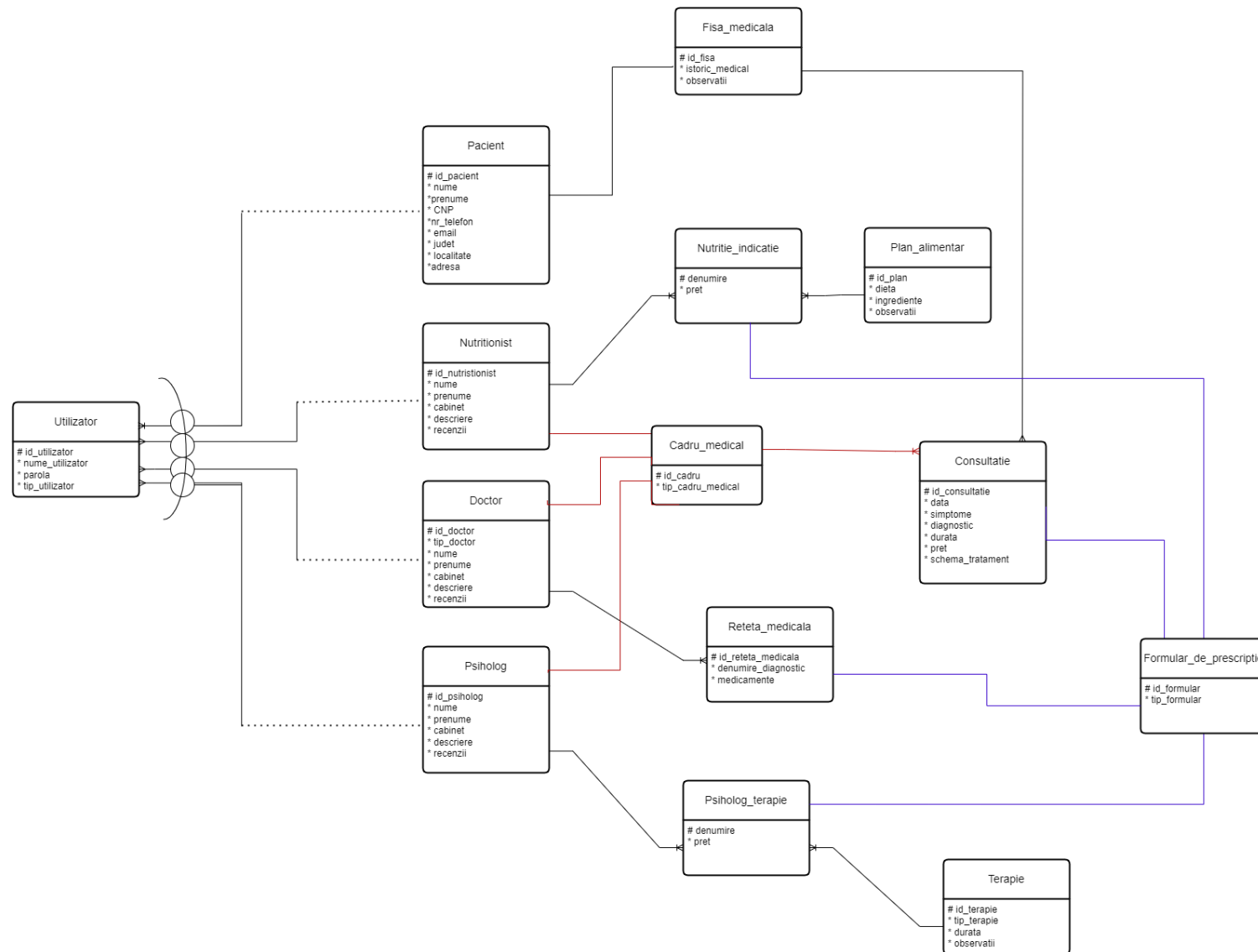


Figura 10. Diagrama entităților și relațiilor

Diagrama entităților și relațiilor (ERD) este utilizată pentru a aduce în prim plan structura și relațiile dintre entități, fiind utilizată în modelarea bazelor de date. Am folosit diagram ERD pentru a reprezenta grafic modelul rațional. În partea de mai jos am prezentat diagrama, pentru a ilustra baza de date cu relațiile sale.

3.1.1 Arhitectura sistemului

Arhitectura clinicii medicale de oncologie este destul de ușoară, împărțită pe 3 nivele: back-end, front-end și baza de date. Această structura oferă date client-server, care argumentează ipoteza de a dezvolta și menține module independente pe platforme separate, următoarele lucruri: logica procesului funcțional, accesul la date, stocarea datelor la calculator și interfața utilizatorului. (Science, 2023)

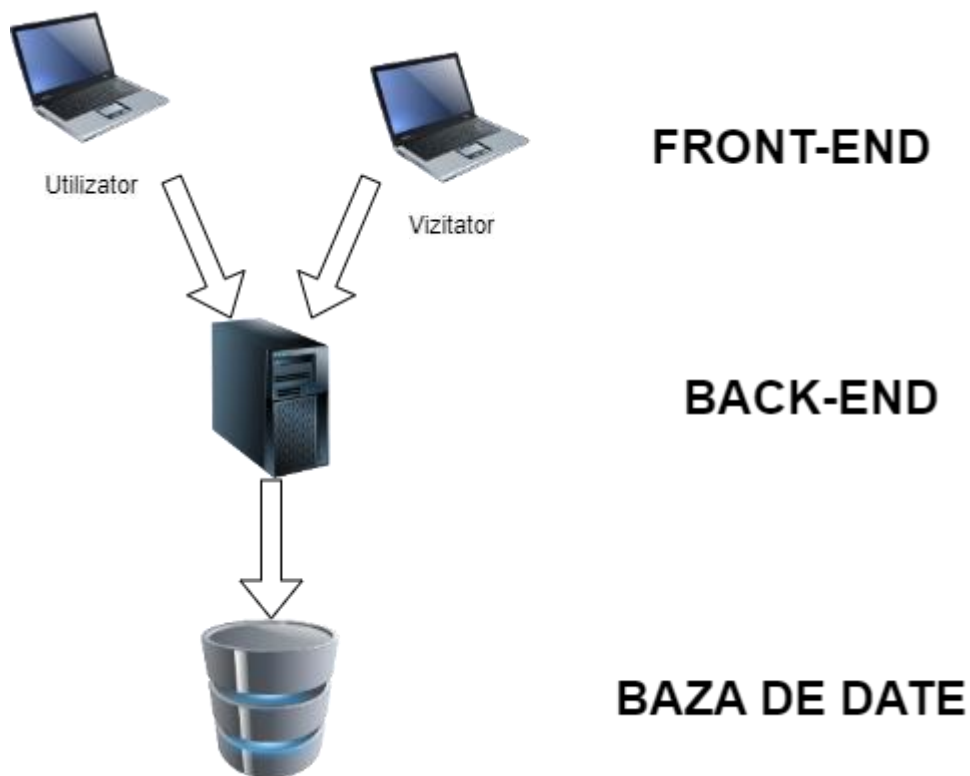


Figura 11. Model arhitectură de sistem pe 3 niveluri

Secvența de front-end evidențiază componentele vizuale ale aplicației cu care interacționează utilizatorii și vizitatorii. Pentru a pătrunde în platforma web, trebuie să aveți conexiune la internet și acces la un browser. Partea de front-end trebuie să fie una lizibilă și cât mai funcțională, dar și cât mai ușor de manevrat, deoarece acest lucru va atrage un număr mare de vizitatori, care pe urmă vor deveni pacienți aplicației. Aceștia vor întâmpina o experiență

plăcută, din contră dacă nu ar fi așa vom pierde un numărul de persoane care ar putea folosi aplicația, așadar trebuie să fie de un rafinament puternic.

Secvența de back-end reprezintă gestionarea conținutului nevăzut de către utilizator, deoarece acesta este desemnat a fi partea de cod din spatele părții vizuale. Cu alte cuvinte, partea de back-end face posibilă interacțiunea dintre aplicație și utilizator, astfel controlând logica platformei web. Cel mai important rol susținut de partea de back-end este conectarea sa la baza de date, pentru a aduce informații eligibile în locurile unde va fi nevoie.

Secvența de baze de date stochează toate informațiile într-un singur loc, iar când avem nevoie de ele, aceasta este apleată și trimite datele acolo unde este nevoie. Pentru o funcționare cât mai performantă a site-ului, trebuie să avem o bază de date bine organizată și structurată. Cum am menționat și anterior, partea de back-end este în strânsă corelare cu baza de date. Un factor important este acela ca din back-end noi putem face diferite operații în baza de date, cum ar fi: vizualizare, adăugare, modificare, ștergere.

Deoarece am vrut să prezentăm mai detaliat diagrama pe trei niveluri, am creat o diagramă care ilustrează rolurile fiecărei părți menționate mai sus.

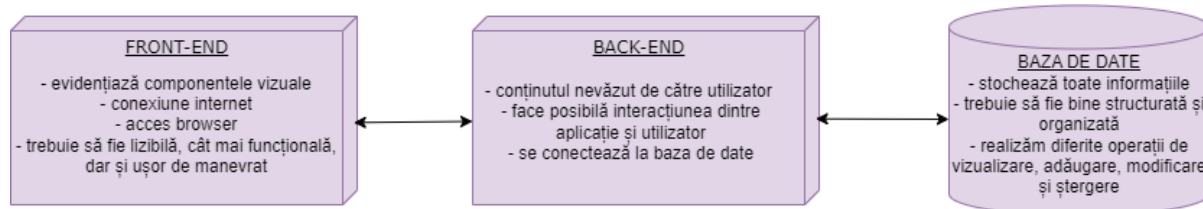


Figura 12. Diagrama pe trei niveluri detaliată

Diagrama de componente simbolizează începutul unui nou proiect, schița acestuia. Aceasta înfățișează conexiunea dintre componentele software logice și relația cu sistemul. Prin această diagramă putem verifica dacă funcționalitatea sistemului este puternică și bună de implementat. În Anexe se va regăsi diagrama de componente întregă, cât timp aici este explicată partea de Psihologi.

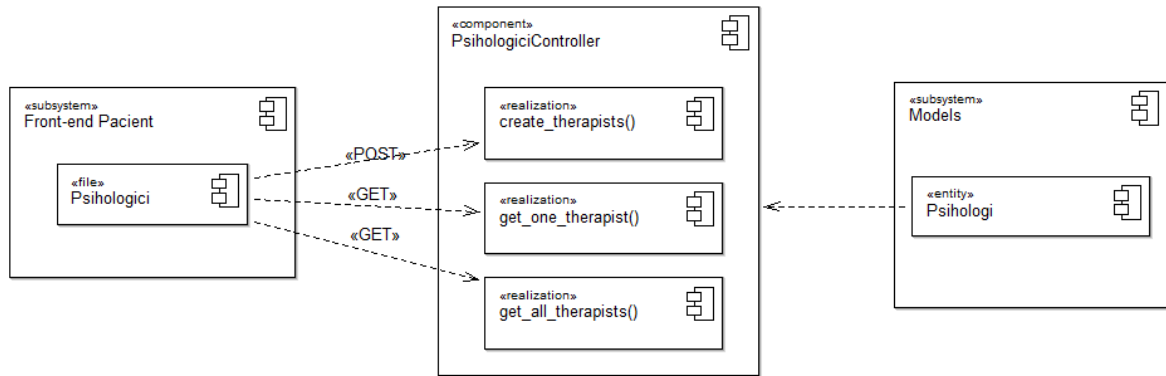


Figura 13. Diagrama de componente

Diagrama deployment este o reprezentare grafică a componentelor software și hardware într-un sistem. Aceasta are componenete și roluri care sunt explicate în diagramă. În diagrama de mai jos am realizat infrastructura proiectului, cu parte de front-end, back-end și baze de date. Mai mult, pentru a o face mai complex am pus și câteva librării pe care le-am folosit.

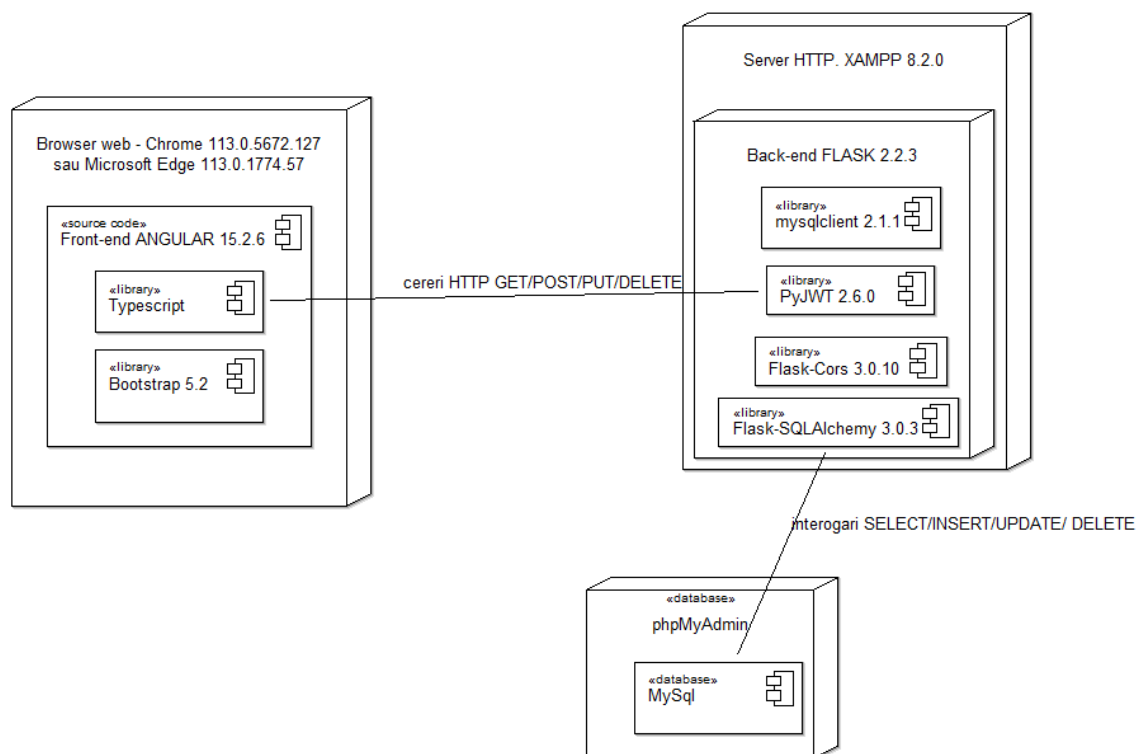


Figura 14. Diagrama deployment

3.1.2 Baza informațională

Următoarea componentă majoră a oricărei aplicații web este sistemul informatic, mai exact baza de informații. Practic înseamnă ca datele sunt prelucrate, având un flux de date și sistem. Prin platforma noastră dorim să arătăm un sistem care dorește să ajute lumea prin intermediul cadrelor medicale. Pentru a îndeplini aceste sarcini pe care le-am identificat la capitolul anterior (2.3.1 Documentarea cerințelor). Mai mult, tot acolo am identificat și actorii care aparțin sistemului: vizitatori și utilizatori. Pentru clinica medical de oncologie baza informațională este dăruită de sistemul de gestiune a bazelor de date MySQL. Am creat câteva categorii pentru a exemplifica datele de intrare și ieșire. Acestea sunt următoarele:

Autentificare utilizator – pacient (nume utilizator, parola)

Programare cadrul medical (simptome, data)

Vizualizare cadrul medical (nume, prenume, contact, cabinet, descriere, recenzii)

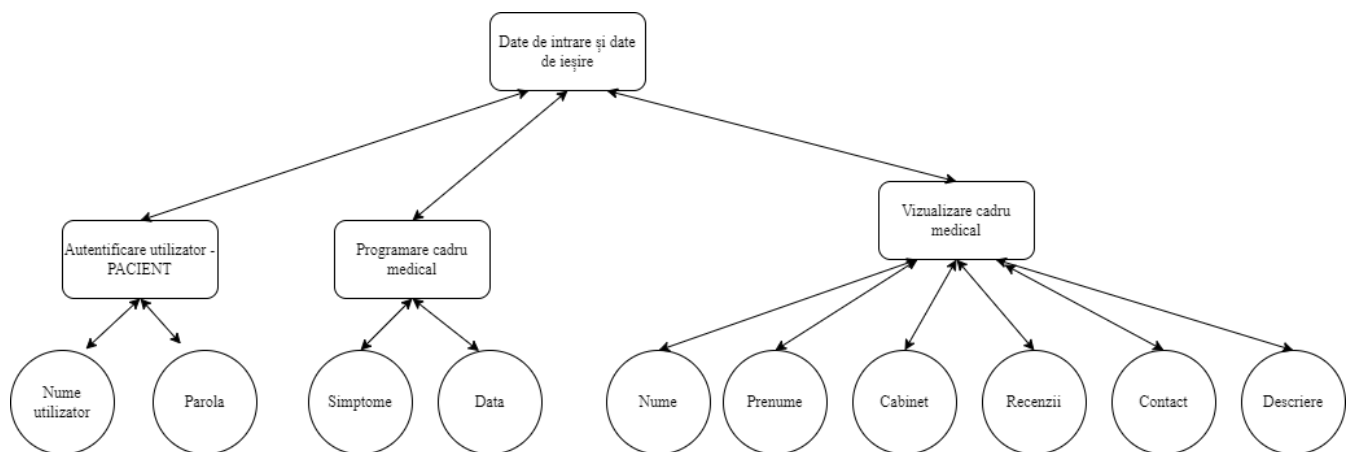


Figura 15. Diagrama de date

3.2 Proiectare tehnică

3.2.1 Structura fizică a datelor

Datele utilizate de sistemul informatic sunt stocate într-o bază de date relațională care se definește ca fiind un tip de bază de date care stochează și permite accesul punctelor de date care au o conexiune între ele. Modelul relațional reprezintă fundamentul pentru gestionarea datelor într-un mod relațional, iar fiecare înregistrare dintr-un tabel are o cheie primară care se caracterizează ca fiind un cod unic. De asemenea, coloanele din tabel poartă denumirea de

atribute, iar fiecare înregistrare trebuie să aibă un atribut, ce facilitează stabilirea relațiilor între punctele de date. (Oracle, 2023)

phpMyAdmin este o aplicație software gratuită, concepută pentru a administra bazele de date MySQL prin intermediul unei interfețe web. Permite accesul la o gamă largă de operații MySQL, cât și MariaDB. Există posibilitatea de a se executa direct instrucțiuni SQL sau prin intermediul interfeței cu utilizatorul se pot realiza diferite operații, cum ar fi gestionarea bazelor de date, tabelelor, coloanelor, relațiilor, indexelor, utilizatorilor, permisiunilor. (phpmyadmin, 2003-2023) Pentru lucrarea noastră am folosit baza de date MySQL. Mai jos am atașat structura tabelelor din designer-ul din phpMyAdmin.

DIGITALIZAREA FLUXULUI DE LUCRU DINTR-O CLINICĂ DE ONCOLOGIE

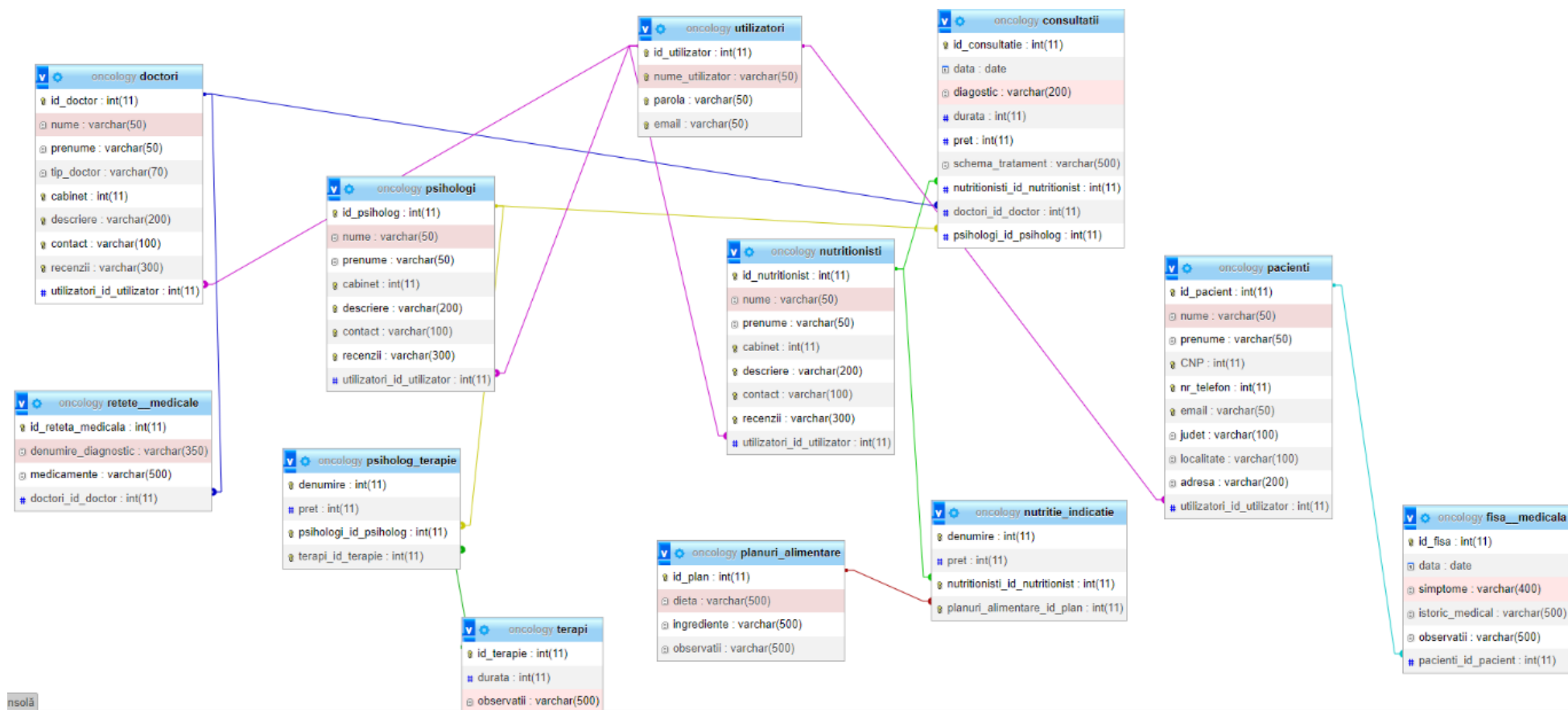


Figura 16. Structura tabelor din designer din phpMyAdmin

3.2.2 Procese și algoritmi

În acest capitol ne-am propus să prezentăm diferite structuri de algoritmi, ușori de interpretat, pe care îi poate înțelege orice persoană cu cunoștințe intermediare de programare. Aplicația web este clară și simplă, așadar cu trecerea timpului orice persoană care va fi responsabilă de întreținere și diverse îmbunătățiri poate face cu ușurință modificările necesare.

Procesul de ÎNREGISTRARE

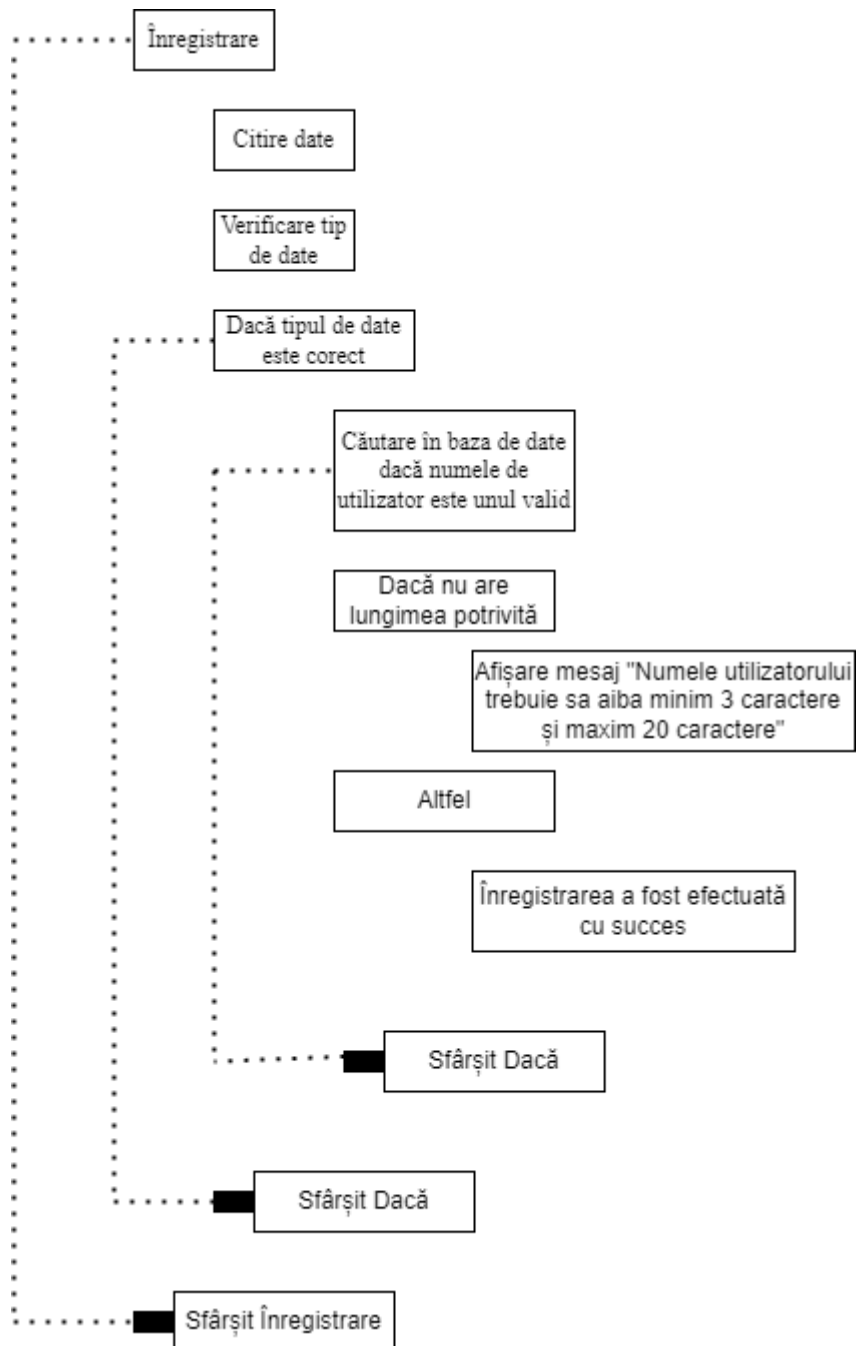


Figura 17. Modelul de înregistrare explicat

Procesul de AUTENTIFICARE

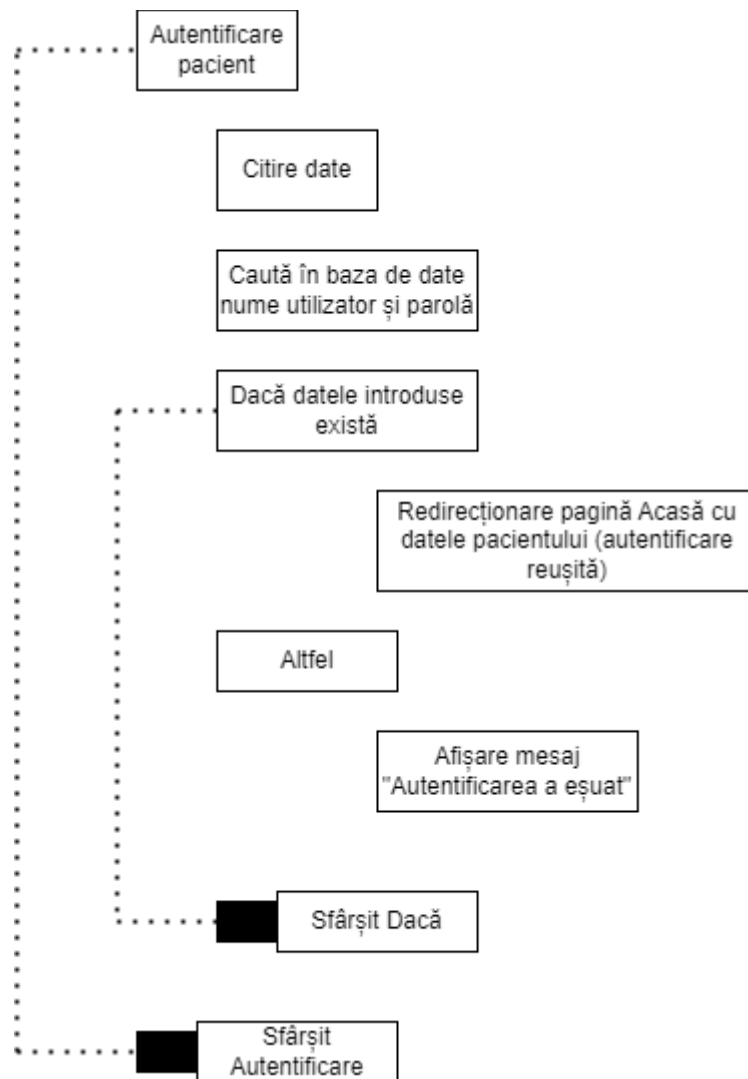


Figura 18. Modelul de autentificare explicat

3.2.3 Tehnologii specifice

În următorul capitol vom descrie tehnologiile folosite, limbajele de programare, dar și cele mai importante librării pentru implementarea aplicației web, clinică medicală cu specific de oncologie.

Componenta BACK-END

Pentru partea de back-end am folosit Python, implementat în framework-ul Flask. L-am folosit pentru gestionarea logicii și a funcționalității aplicației, pentru prelucrarea din baza de date și frunizarea lor către partea de front-end. "Flask este un framework ușor pentru aplicații

web WSGI (Web Server Gateway Interface). A fost conceput pentru a facilita începerea rapidă și ușoară a dezvoltării aplicațiilor, având capacitatea de a se scala și a aborda aplicații complexe. Inițial, Flask era doar un înveliș simplu pentru Werkzeug și Jinja, dar a devenit unul dintre cele mai populare framework-uri pentru aplicații web în limbajul Python. Flask oferă sugestii și recomandări, dar nu impune dependențe sau o structură specifică de proiect. Dezvoltatorul are libertatea de a alege instrumentele și bibliotecile pe care dorește să le utilizeze. Există numeroase extensii oferite de comunitate care facilitează adăugarea de noi funcționalități în aplicație. (Pallets, Flask, n.d.) Flask este automat pus pe portul 5000.

Primul pas a fost să instalăm ultima versiune a lui Python 3.11.2. Automat când am instalat Flask 2.2.3, s-a implementat și Werkzeug (WSGI) cu aceeași versiune care semnifică interfața standard între aplicații și server. Pentru a eficientiza proiectul, am creat un mediu virtual de lucru. Pe viitor este de folos, deoarece dacă dezvoltatorul dorește să conceapă mai multe proiecte Python, cu librării diferite sau versiuni diferite, acestea nu se vor încurca și nu vor genera erori. După instalarea mediului de lucru, de fiecare dată când vrem să-l deschidem vom pune în terminal comanda **env\Scripts\activate** pentru a porni mediul. Pentru a rula aplicația vom folosi sintaxa: **python -m flask run**. (Pallets, Installation, 2010)

Componenta FRONT-END

Partea de front-end este realizată în Angular, iar aceasta este o platformă de dezvoltare, care pune la dispoziție limbajul TypeScript. Angular oferă diferite componente pentru construirea aplicațiilor web, cum ar fi o colecție de biblioteci care acoperă o varietate mare de caracteristici (rutare, gestionarea formularelor, comunicarea client-server și multe altele). Instalarea limbajului se face în linia de comandă, acolo unde se va face și compilarea acestuia. Angular este pus pe portul 4200, iar pentru a rula programul se introduce în cmd sintaxa **ng serve --open** care se va deschide automat aplicația într-o pagină nouă de browser.

„Aplicația este compusă din blocuri de componente. Aceasta include o clasă TypeScript cu un decorator **@Component()**, un șablon HTML și stiluri CSS. Pe de o parte, CSS se definește ca un limbaj descriptiv care se folosește pentru descrierea semanticii de prezentare a unui alt document scris în limbaj descriptiv. Un selector CSS definește modul în care componenta este utilizată într-un șablon. Pe de altă parte, HTML e un limbaj independent de platformă folosit pentru a structura și descrie conținutul diferitelor documente, inclusiv pagini web. Cu ajutorul acestuia, setăm elemente specifice, definite în mod specific, numite etichete și le folosim pentru

a ne defini structura. Elementele HTML din șablon devin instanțe ale componentei. „ (Angular, 2010-2023)

“Angular este un framework și o platformă deschisă pentru crearea de aplicații cu o singură pagină (Single Page Applications), scris în TypeScript, susținut și dezvoltat de către Google. Angular inițial era planificat să fie versiunea 2 a popularului framework AngularJS. Angular prezintă avantaje evidente ca framework, oferind în același timp o structură standard pentru dezvoltatori. Acesta permite utilizatorilor să creeze aplicații mari într-un mod ușor de întreținut.” (Koffer, 2022)

4. Implementarea sistemului informatic

4.1 Mediul de programare

Editorul în care am creat aplicația web se numește Visual Studio Code, care este un IDE (integrated development environment) dezvoltat de Microsoft. Acesta este un editor de cod care rulează pe desktop și l-am folosit pe sistemul de operare Windows. (Microsoft, 2023) „IDE-ul este un mediu de dezvoltare integrate care permite dezvoltatorilor să scrie diferite programe de calculator. În plus, un IDE este o alegere favorabilă pentru programator, deoarece poate combina activitățile de scriere a software-ului într-o singură aplicație, cum ar fi editarea codului sursă, construirea de executabile și depanarea.” (Team, 2023)

Ambele părți, front-end și back-end, le-am realizat în Visual Studio Code. Am instalat diferite extensii pentru a ne simplifica munca, iar câteva dintre acestea sunt: “Angular Language Service”, “Angular Essentials (Version 13)”, “npm”, “Python” și “Pylance”. După cum urmează vom prezenta modul în care a prins viață clinica medicală de oncologie, ajutându-ne de zone de front-end, back-end și baza de date.

4.2.1 Pregătirea bazei de date

Pentru început am instalat Xampp Control Panel, deoarece aveam nevoie de porturile puse la dispoziție, mai exact Apache, care rulează pe portul 80, 443 și MySQL care e pe portul 3306. Am ales acest mediu de dezvoltare, întrucât l-am folosit la facultate și aveam cunoștințe legate de el. Am decis ca baza de date să fie una relațională și pe care să o putem gestiona din interfața phpMyAdmin.

Primul pas parcurs în realizarea aplicației a fost să conectăm back-end-ul la baza de date. În primul rând, am instalat următoarele librării, deoarece au funcții specifice în contextul dezvoltării aplicațiilor web cu Flask și interacțiunii cu baza de date MySQL.

Flask-SQLAlchemy 3.0.3 este o extensie Flask care facilitează integrarea și utilizarea bibliotecii SQLAlchemy în aplicația Flask. Mai mult, librăria permite definirea unui model de

date Python care mapează tabelele din baza de date MySQL și utilizează metode și interogări SQLAlchemy pentru a citi, scrie și actualiza datele.

Mysql-connector-python 8.0.32 și **mysqlclient 2.1.1** sunt două biblioteci Python folosite pentru interacțiunea cu baza de date MySQL. Prima este librăria oficială Python dezvoltată de MySQL. Cele două oferă o interfață eficientă pentru realizarea conexiunii, dar și pentru a interoga și actualiza datele. Mysqlclient este recunoscut pentru o performanță mare, de aceea este foarte folosită în proiectele în care e nevoie de o interacțiune rapidă cu bazele de date MySQL.

SQLAlchemy 2.0.9 facilitează crearea și editarea modelelor de date, gestionarea relațiilor dintre tabel și rularea interogărilor complexe. Este o bibliotecă Python puternică și flexibilă cu bazele de date relaționale.

Pasul următor pentru realizarea conectării a fost să chemăm baza de date în back-end, unde am atașat mai jos codul aferent.

```
app.config['SECRET_KEY'] = 'thisissecret'
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:' '@localhost/oncologie'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db.init_app(app)
```

Figura 19. Conectare la baza de date

În interfața de phpMyAdmin vom vedea și putea edita tabelele create cât și conținutul lor. Fiind un mediu de dezvoltare optim, phpMyAdmin ușurează munca prin organizarea tabelor și a conținutului lor. Pentru început am creat baza de date care poartă numele de **ONCOLOGIE**. Tabelele le-am implementat din back-end, de asemenea, unele înregistrări s-au realizat din Postman sau chiar din phpMyAdmin.

Tabel	Acțiune	Rânduri	Tip	Coloțonare	Dimensiune	Overhead
<input type="checkbox"/> cadre_medicale	★ Răstoiește Structură Caută Inserează Golește Elimină	3	InnoDB	utf8mb4_general_ci	16,0 K10	-
<input type="checkbox"/> consultatii	★ Răstoiește Structură Caută Inserează Golește Elimină	15	InnoDB	utf8mb4_general_ci	64,0 K10	-
<input type="checkbox"/> doctori	★ Răstoiește Structură Caută Inserează Golește Elimină	6	InnoDB	utf8mb4_general_ci	64,0 K10	-
<input type="checkbox"/> fisa_medicala	★ Răstoiește Structură Caută Inserează Golește Elimină	6	InnoDB	utf8mb4_general_ci	32,0 K10	-
<input type="checkbox"/> formular_de_prescriptie	★ Răstoiește Structură Caută Inserează Golește Elimină	4	InnoDB	utf8mb4_general_ci	16,0 K10	-
<input type="checkbox"/> nutritie_indicatie	★ Răstoiește Structură Caută Inserează Golește Elimină	1	InnoDB	utf8mb4_general_ci	64,0 K10	-
<input type="checkbox"/> nutritionisti	★ Răstoiește Structură Caută Inserează Golește Elimină	2	InnoDB	utf8mb4_general_ci	48,0 K10	-
<input type="checkbox"/> pacienti	★ Răstoiește Structură Caută Inserează Golește Elimină	9	InnoDB	utf8mb4_general_ci	64,0 K10	-
<input type="checkbox"/> planuri_alimentare	★ Răstoiește Structură Caută Inserează Golește Elimină	1	InnoDB	utf8mb4_general_ci	16,0 K10	-
<input type="checkbox"/> psihologi	★ Răstoiește Structură Caută Inserează Golește Elimină	4	InnoDB	utf8mb4_general_ci	64,0 K10	-
<input type="checkbox"/> psiholog_terapie	★ Răstoiește Structură Caută Inserează Golește Elimină	1	InnoDB	utf8mb4_general_ci	64,0 K10	-
<input type="checkbox"/> retete_medicale	★ Răstoiește Structură Caută Inserează Golește Elimină	2	InnoDB	utf8mb4_general_ci	48,0 K10	-
<input type="checkbox"/> terapii	★ Răstoiește Structură Caută Inserează Golește Elimină	1	InnoDB	utf8mb4_general_ci	16,0 K10	-
<input type="checkbox"/> utilizatori	★ Răstoiește Structură Caută Inserează Golește Elimină	16	InnoDB	utf8mb4_general_ci	32,0 K10	-
14 tabele	Total	71	InnoDB	utf8mb4_general_ci	608,0 K10	0 0

Figura 20. Vizualizare tabele din phpMyAdmin

4.2.2 Pregătirea zonei de back-end

După cum am explicat și în capitolul precedent, 3.2.3 Tehnologii specifice, am instalat limbajul de programare Python și framework-ul web Flask. După ce am realizat importurile necesare unei bune funcționări, am generat un nou fișier, **app.py** care cuprinde inițializarea framework-ului Flask, urmând documentația oficială:

```
from flask_cors import CORS
from flask import Flask, request, jsonify, make_response
from flask_sqlalchemy import SQLAlchemy
from flask_marshmallow import Marshmallow
from models import db, Doctori, Utilizatori, Pacienti, Nutritionisti,
Psihologi, Fisa_Medicala, Planuri_Alimentare, Consultatii, Retete_Medicale,
Terapii, Formular_de_prescriptie, Cadre_medicale, Nutritie_Indicatie,
Psiholog_Terapie
import uuid
import jwt
import datetime
from functools import wraps

app = Flask(__name__)
CORS(app, supports_credentials=True)

app.config['SECRET_KEY'] = 'thisissecret'
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:' + '@localhost/oncologie'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db.init_app(app)
```

Figura 21. Vizualizare inițializare Flask

Pe urmă am creat tabelele în fișier **models.py** care le construiește în baza de date MySQL, mai exact în platforma phpMyAdmin. Mai jos voi atașa codul corespunzător creării tabelului Utilizatori.

```
class Utilizatori(db.Model):
    __tablename__ = 'utilizatori'

    id_utilizator = db.Column(db.Integer, primary_key = True)
    nume_utilizator = db.Column(db.String(50), nullable = False)
    parola = db.Column(db.String(50), nullable = False)
    email = db.Column(db.String(50), nullable = False)
    tip_utilizator = db.Column(db.String(50), nullable = False)
    public_id = db.Column(db.String(50), unique = True)
    admin = db.Column(db.Boolean)
    pacienti = db.relationship('Pacienti', backref='utilizatori', uselist=False)
    nutritionisti = db.relationship('Nutritionisti', backref='utilizatori',
uselist=False)
    psihologi = db.relationship('Psihologi', backref='utilizatori',
uselist=False)
    doctori = db.relationship('Doctori', backref='utilizatori', uselist=False)

    def __init__(self, nume_utilizator, parola, email, tip_utilizator,
public_id, admin):
        self.nume_utilizator = nume_utilizator
        self.parola = parola
        self.email = email
        self.tip_utilizator = tip_utilizator
        self.public_id = public_id
        self.admin = admin
```

Figura 22. Vizualizare creare tabel Utilizatori

“Protocolul de transfer hipertext (HTTP) este protocolul folosit de programe pentru a comunica prin World Wide Web. HTTP este cel mai faimos pentru conversația bidirecțională între browsere web și servere web.” (Gourley, Totty, Sayer, Aggarwal, & Reddy, 2002)

“Cererile HTTP sunt de mai multe tipuri (metode), dintre care unele au fost evidențiate deja la discutarea tehnicii REST (acolo aceste tipuri de cereri sunt tratate ca interogari JSON):

- Cererile GET de regulă solicită date de la un server
- Cererile POST de regula trimit date spre server în scopul stocării lor acolo
- Cererile PUT trimit date spre server în scopul înlocuirii unora existente acolo

- Cererile DELETE solicită ștergerea unor date de pe server” (Buchmann & Ghiran, 2023)

După ce am finalizat construcția bazei de date, am creat API-urile corespunzătoare clinicii medicale. Vom da exemple pentru fiecare metodă HTTP folosită.

Am implementat mai multe rute **POST** pentru a realiza înregistrări în cadrul bazei de date. Mai jos prezentăm endpoint-ul creat pentru tabela Nutriționiști, care este susținut de adresa *'/nutritionisti'*.

```
@app.route('/nutritionisti', methods=['POST'])
@token_required
def create_nutritionist(current_user):

    data = request.get_json()

    cadru_medical = Cadre_medicale(tip_cadru_medical="nutritionist")

    try:
        db.session.add(cadru_medical)
        db.session.commit()
        nutritionist = Nutritionisti(nume=data['nume'], prenume=data['prenume'],
cabinet=data['cabinet'], descriere=data['descriere'], contact=data['contact'],
recenzii=data['recenzii'], utilizatori_id_utilizator=data['utilizatori_id'],
cadre_medicale_id_cadru=cadru_medical.id_cadru)

        try:
            db.session.add(nutritionist)
            db.session.commit()
            return jsonify({'mesaj': 'Nutritionist adăugat cu succes!'})
        except:
            return jsonify({'mesaj': 'Eroare la adăugarea nutritionistului!'}),
500
    except:
        return jsonify({'mesaj': 'Eroare la adăugarea cadrului medical!'}),
500
```

Figura 23. Creare metodă POST, pentru tabela Nutriționiști

Am creat rute **GET** pentru a aduce în interfața aplicației tot ce ne pune la dispoziție tabela responsabilă creării metodei respective. Mai mult, pentru a avea mai multe detalii, dar mai ales pentru a putea avea în aplicația noastră doar câte un atribut chemat din tabelă, am efectuat ruta *GET/id*.

```

@app.route('/planuri_alimentare', methods=['GET'])
@token_required
def get_all_diet_plan(current_user):

    global Planuri_Alimentare

    diet_plans = Planuri_Alimentare.query.all()

    output = []

    for Planuri_Alimentare in diet_plans:
        diet_plans_data = {}
        diet_plans_data['dieta'] = Planuri_Alimentare.dieta
        diet_plans_data['ingrediente'] = Planuri_Alimentare.ingrediente
        diet_plans_data['observatii'] = Planuri_Alimentare.observatii
        output.append(diet_plans_data)

    return jsonify({'diet_plans' : output})

```

Figura 24. Creare metodă GET, pentru tabela Planuri alimentare

```

@app.route('/planuri_alimentare/<id_plan>', methods=['GET'])
@token_required
def get_one_diet_plan(current_user, id_plan):

    global Planuri_Alimentare

    diet_plan = Planuri_Alimentare.query.filter_by(id_plan = id_plan).first()

    diet_plan_data = {}
    diet_plan_data['dieta'] = diet_plan.dieta
    diet_plan_data['ingrediente'] = diet_plan.ingrediente
    diet_plan_data['observatii'] = diet_plan.observatii

    return jsonify({'diet_plan' : diet_plan_data})

```

Figura 25. Creare metodă GET/id, pentru tabela Planuri alimentare

Am aplicat metoda **DELETE**, pentru tabela Utilizatori pentru a realiza o ștergere în Postman a unui utilizator nedorit sau care a fost creat greșit.

```

@app.route('/utilizatori/<public_id>', methods = ['DELETE'])
@token_required
def delete_user(current_user, public_id):

```

```

if not current_user.admin:
    return jsonify({'message' : 'Cannot perform that function!'})

user = Utilizatori.query.filter_by(public_id = public_id).first()

if not user:
    return jsonify({'message' : 'No user found!'})

db.session.delete(user)
db.session.commit()

return jsonify({'message' : 'Utilizatorul a fost sters!'})

```

Figura 26. Creare metodă DELETE, pentru tabela Utilizatori/public_id

În ultimul rând, în aplicația noastră cadrul medical va putea modifica consultația, adăugând următoarele atribute, care aparțin de tabela Consultații: schema_tratament, diagnostic, durată, preț, formular_de_prescriptie_id_formular. Astfel am creat o metodă **PUT**.

```

@app.route('/consultatii/<id_consultatie>', methods = ['PUT'])
@token_required
def update_consultation(current_user, id_consultatie):

    consultation = Consultatii.query.get(id_consultatie)

    if not consultation:
        return jsonify({'message' : 'Nicio consultatie nu a fost gasita!'})

    diagnostic = request.json['diagnostic']
    schema_tratament = request.json['schema_tratament']
    durata = request.json['durata']
    pret = request.json['pret']
    formular_de_prescriptie_id_formular =
request.json['formular_de_prescriptie_id_formular']

    consultation.diagnostic = diagnostic
    consultation.schema_tratament = schema_tratament
    consultation.durata = durata
    consultation.pret = pret
    consultation.formular_de_prescriptie_id_formular =
formular_de_prescriptie_id_formular

    db.session.commit()

    return jsonify({'message' : 'Actualizarea efectuată de cadrul medical!'})

```

Figura 27. Creare metodă PUT, pentru tabela Consultații/id_consultație

Datorită API-ului am creat un decorator personalizat care are rolul de a impune autorizare, solicitând utilizatorului să genereze un jeton de autentificare (token), furnizat de librăria JWT. Acesta va permite accesul în aplicație, pentru a realiza diferite operațiuni, doar persoanelor înregistrate și autentificate, de aceea am adăugat decoratorul la toate metodele.

```
def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = None

        if 'x-access-token' in request.headers:
            token = request.headers['x-access-token']

        if not token:
            return jsonify({'message' : 'Token is missing!'}), 401
        try:
            data = jwt.decode(token, app.config['SECRET_KEY'],
                                algorithms=['HS256'])
            current_user =
Utilizatori.query.filter_by(id_utilizator=data['user_id']).first()
        except:
            return jsonify({'message' : 'Token invalid!'}), 401

        return f(current_user, *args, **kwargs)

    return decorated
```

Figura 28. Definire jeton de autentificare (token)

4.2.3 Pregătirea zonei de front-end

Pentru pregătirea zonei de front-end ne-am făcut o idee în ansamblu despre ce vom avea de realizat, datorită documentației oficiale Angular.

Am introdus în terminal comanda specifică instalării limbajului, **npm install -g @angular/cli**. Am creat un spațiu de lucru prin comanda **ng new**, care a solicitat informații despre caracteristicile incluse în aplicație, pe care le-am acceptat. După ce am respectat pașii de inițializare a aplicației, am compilat programul și am primit ca rezultat următoarea pagina:

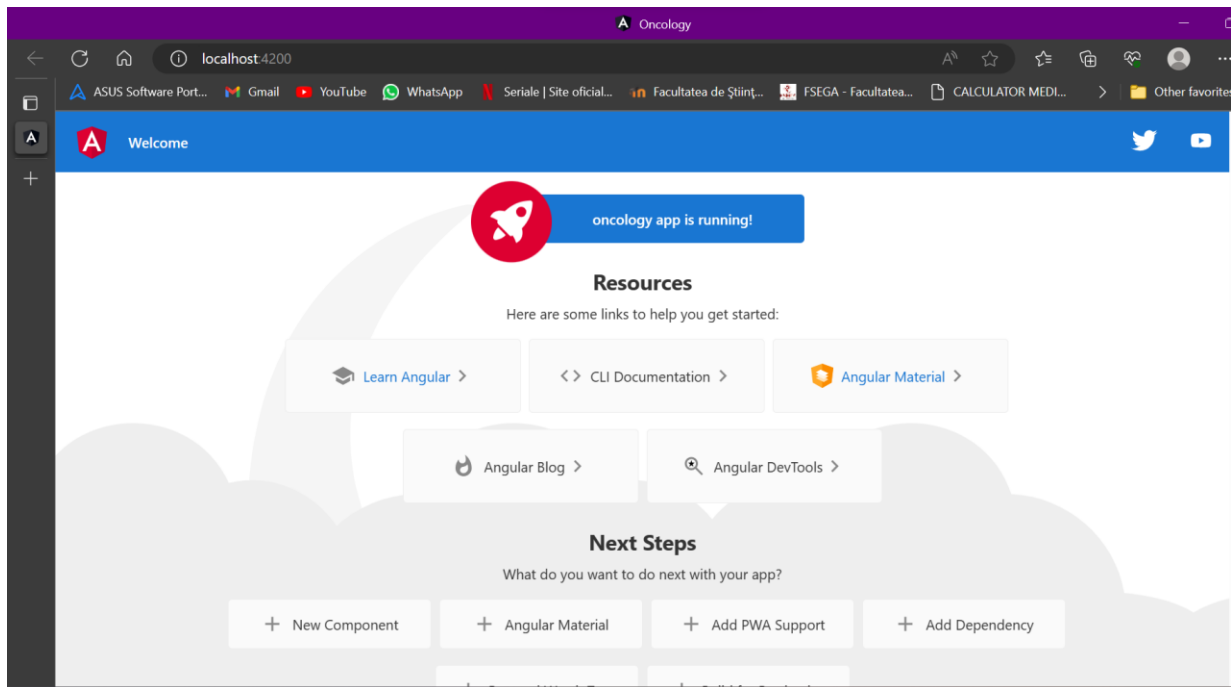


Figura 29. Pagină inițială Angular

Zona de front-end este amplu dezvoltată, având diverse componente care să aducă aplicația în îndeplinirea obiectivelor propuse la început. În folder-ul **src/assets** am furnizat toate imaginile pe care le dorim în platforma web create. În **src/app** s-a lucrat aplicația, unde am creat diverse fișiere. Scopul fișierului **_services** este de a furniza funcționalități și date, într-un mod organizat și eficient, care pot fi partajate între mai multe componente.

Fișierul **app-routing.module** deservește la gestionarea rutelor și la navigarea între diferitele componenete create în Angular. Un alt rol pe care îl îndeplinește este de a defini și configura rutele aplicației web, precum și comportamentul lor asociat cu fiecare rută.

```
const routes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'login', component: LoginComponent },
  { path: 'register', component: RegisterComponent },
  { path: 'utilizator', component: UtilizatoriComponent },
  { path: 'psiholog', component: PsihologiComponent },
  { path: 'nutritionist', component: NutritionistiComponent },
  { path: 'mediconcolog', component: MediconcologComponent },
  { path: 'chirurgoncolog', component: ChirurgoncologComponent },
  { path: 'hematooncolog', component: HematooncologComponent },
  { path: 'oncologmedical', component: OncologmedicalComponent },
  { path: 'oncologradioterapeut', component: OncologradioterapeutComponent },
  { path: 'adaugare.consultatie/:id', component: AdaugareConsultatieComponent },
],
{ path: 'consultatii', component: ConsultatiiComponent },
```

```

{ path: 'programarile.mele', component: ProgramarileMeleComponent },
{ path: 'adaugare.consultatie.psiholog/:id', component:
AdaugareConsultatiePsihologComponent },
{ path: 'adaugare.consultatie.nutritionist/:id', component:
AdaugareConsultatieNutritionistComponent },
{ path: 'istoric.medical', component: IstoricMedicalComponent },
{ path: 'planuri.alimentare', component: PlanuriAlimentareComponent },
{ path: 'terapii', component: TerapiiComponent },

```

Figura 30. Rutele paginilor web

Fișier **app.module.ts** reprezintă modulul rădăcină al aplicației Angular, unde sunt importate și configurate diverse module, componente, servicii și alte resurse necesare pentru a rula aplicația. Aceasta acționează ca punct de intrare principal pentru platforma și este responsabil pentru încărcarea și pornirea AppComponent și orice alte componente sau module necesare aplicației.

Am chemat endpoint-urile definite în back-end, deoarece evidențiază relația de client-server. Voi lăsa mai jos un exemplu pentru o rută GET și POST apleată în Angular pentru procesul de înregistrare și autentificare.

```

login(num_utilizator: string, parola: string): Observable<any> {
  const str = num_utilizator + ":" + parola;
  const response = this.http.get(
    AUTH_API + 'login',
    {responseType: 'json', headers: new HttpHeaders({ 'Content-Type':
'application/json', 'Access-Control-Allow-Origin': '*', "Authorization":
"Basic " + btoa(str) })}, withCredentials: true}
  );
  return response;
}

register(num_utilizator: string, parola: string, email: string, nume:
string, prenume: string, CNP: string, nr_telefon: number, judet: string,
localitate: string, adresa: string): Observable<any> {
  return this.http.post(
    AUTH_API + 'pacienti',
    {
      num_utilizator,
      parola,
      email,
      nume,
      prenume,
      CNP,
      nr_telefon,
      judet,
      localitate,

```

```

        adresa,
      },
      httpOptions
    );
  }

```

Figura 31. Vizualizare rute GET și POST

Pentru design am folosit Bootstrap pentru a ușura muncă, dar și a face interfața site-ului să arate nemaipomenit de bine. „Acesta este un software open-source care ajută la dezvoltarea de site-uri web și este disponibil pentru utilizarea de către oricine. Acesta facilitează crearea de site-uri responsiv, oferind o interfață modernă și profesionistă. Recent, Bootstrap a fost actualizat la versiunea Bootstrap 5. În iunie 2020, a fost lansată versiunea alfa Bootstrap 5 după luni de procese de rafinare. Bootstrap este în mod original o combinație de cod CSS și JavaScript pe care îl puteți folosi pentru a crea site-uri web dinamice. Este o unealtă eficientă pentru designul front-end, care include șabloane de design HTML și CSS. Acestea includ componente precum formulare, butoane, navigare, modale, tipografie și alte elemente de interfață. Toate acestea pot fi create utilizând extensii JavaScript.” (Balooni, 2021)

Pagina de Acasă este disponibilă utilizatorilor cu rolul de vizitator. Pentru a arăta cât mai bine am folosit componenta Carousel din Bootstrap. Ne dorim să fie o pagină care atrage mulți utilizatori.

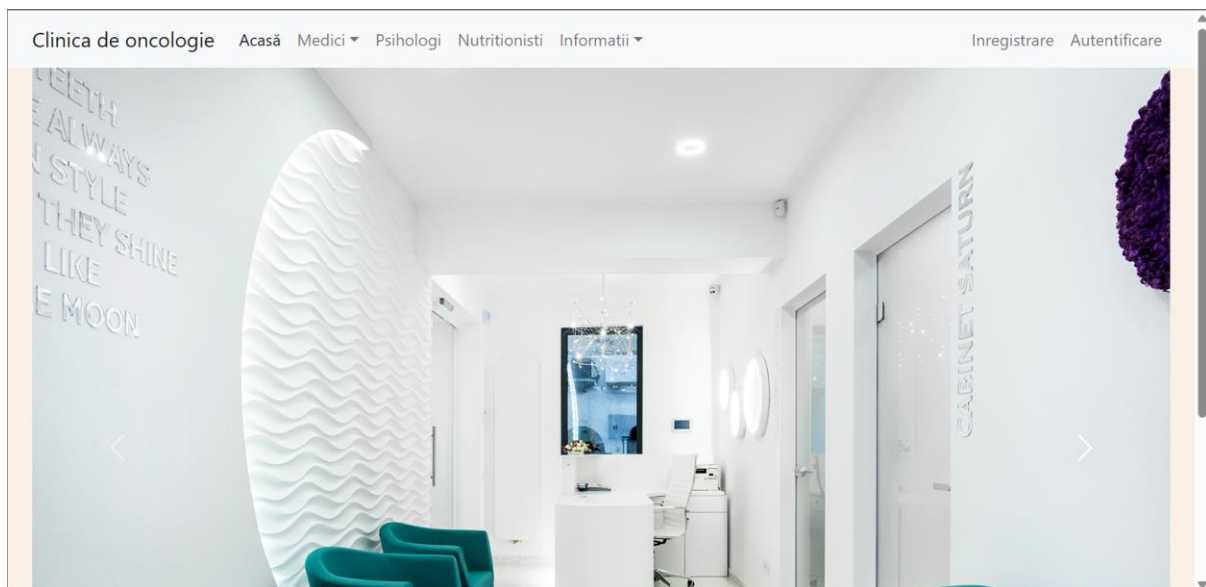


Figura 32. Pagina Acasă - rol vizitator

După ce se autentifică în aplicația web, cadrul medical are o pagină de Acasă personalizată:

Clinica de oncologie Acasă Consultații Listă pacienți Delogare

Bine ai venit, Iancu!

Informații personale

Nume și prenume	Iancu	Doriana
Cabinet	200	

Detalii suplimentare

Descriere: Psihoterapii cognitive si comportamentale. Psihologie clinica si psihoterapie.

Contact: 0793619737

Recenzii: bun

Figura 33. Pagina Acasă - rol cadru medical

Din perspectiva autentificării pacientului, pagina de Acasă se schimbă având Informații personale și Detalii medicale.

Clinica de oncologie Acasă Medici Psihologi Nutritionisti Programările mele Istoric consultații Informații Delogare

Bine ai venit, Pop!

Informații personale

Nume și prenume	Pop	Andrei
CNP	50011090111171	
Număr telefon	798256143	
Email	andrey@yahoo.com	
Județ	Cluj	
Localitate	Gherla	
Adresă	str Lalele 10	

Detalii medicale

Istoric Medical: -

Observații: intoleranță la gluten

Figura 34. Pagina Acasă - rol pacient

Pagina de Consultații este sub responsabilitatea cadrului medical, reprezentând toate programările pe care le are de astăzi și din viitor.

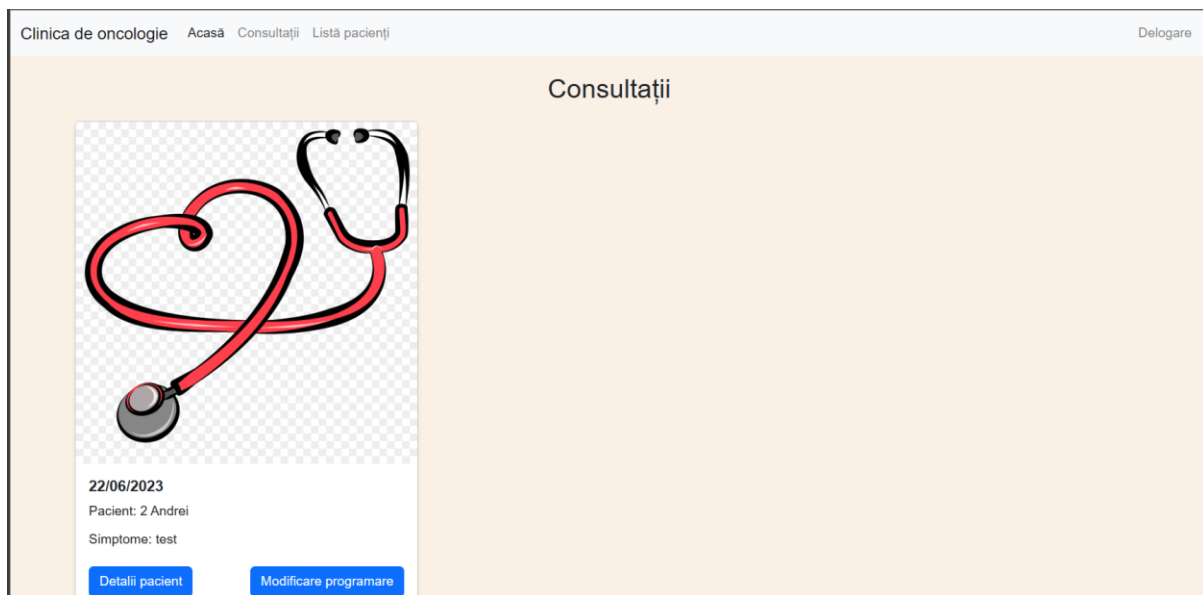


Figura 35. Pagina Consultatii

Pentru vizitatorul, dar și pentru pacient s-a creat o pagină de planuri alimentare care să ofere informații de planurile pe care le poate oferi nutriționistul clinicii.

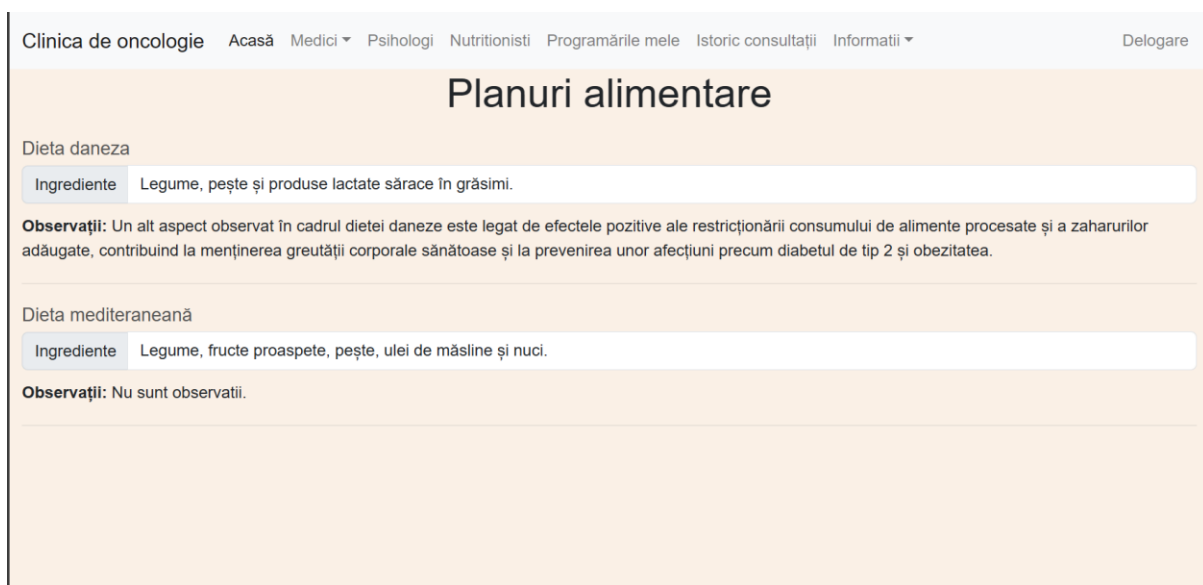


Figura 36. Pagina Planuri alimentare

Clientul are destinată o pagină de Programările mele unde sunt inițializate toate consultările de astăzi și din viitor.

Clinica de oncologie Acasă Medici ▾ Psihologi Nutritionisti Programările mele Istoric consultații Informații ▾ Delogare

Programări de consultare

22/06/2023	23/06/2023	22/06/2023	22/06/2023	22/06/2023
Simptome: test	Simptome: test	Simptome: test	Simptome: test	Simptome: tst
Medic: Popa Bianca	Medic: Sipos Mihai	Medic: Nelus Andrada	Medic: Litan Lavinia	Medic: Iancu Doriana
Cabinet: 303	Cabinet: 115	Cabinet: 110	Cabinet: 204	Cabinet: 200

Figura 37. Pagina Programările mele - pacient

5. Testarea platformei

5.1 Date generale legate de testare

Testarea reprezintă un lucru crucial în dezvoltarea unei aplicații de orice fel. Trebuie realizată pentru a depista diferite erori sau probleme ce pot apărea la conectare, la definirea rutelor. În urma acestui motiv este important să se realizeze teste periodice pentru a furniza o aplicație cât mai performantă. “Următoarele caracteristici reprezintă beneficiile testării:

- Crește gradul de încredere în produsul dezvoltat;
- Crește calitatea produsului;
- Scad costurile de mentenanță;
- Crește performanța produsului;
- Crește gradul de satisfacție al persoanei/persoanelor care au un interes în produs;
- Creșterea profitului.” (Jecan, 2022)

5.1.1 Testarea de utilizabilitate

Pentru testarea de utilizabilitate am apelat la grupul înființat pentru dezvoltarea ideilor și a obiectivelor aplicației web, menționat la subcapitolul 2.2.2 Metoda Brainstorming. Acesta a încercat aplicația și fiecare participant a lăsat câte o recenzie cu privire la experiența avută. Unii subliniază că anumite elemente ar trebui eliminate și altele adăugate pentru a îmbunătăți calitatea generală. Este importantă ascultarea diferitelor opinii legate de un produs, pentru a atinge un potențial cât mai ridicat asupra clinicii medicale. Un exemplu de modificare a fost susținut de către Cristina legat de oferirea unor detalii mai generoase cu privire la descrierea cadrelor medicale, ci să nu mai fie atât de succinte.

5.1.2 Testarea de compatibilitate

Încă de la început, un obiectiv a fost să facem ca aplicația să ruleze pe cât mai multe dispozitive, sau chiar pe cât mai multe sisteme de operare, așa că am simțit că este necesar să testăm pe cât

mai multe în ceea ce privește compatibilitatea. Pentru a testa acest lucru, am rulat mai întâi aplicația pe mai multe browsere pentru a mă asigura că funcționează acolo. Cum ar fi:

- Brave
- Google Chrome
- Microsoft Edge
- Mozilla Firefox
- Opera

Mai apoi, am testat modul în care site-ul web s-a comportat pe diferite dispozitive și diferite versiuni de Windows, de la versiunea 8 până la versiunea 11. După testare, am ajuns la concluzia că aplicația funcționează fără modificări pe majoritatea dispozitivelor.

5.1.3 Testarea de integritate a sistemului

Testarea de integritate a sistemului este rezultatul mai multor teste mai mici care sunt efectuate periodic pentru asigurarea că fiecare caracteristică nouă adăugată la proiect funcționează. Folosim aceste teste mici pentru că simplifică procesul de creare și evităm riscul de a găsi erori critice în părțile mai avansate ale aplicației. Cu alte cuvinte, scopul acestor teste este de a identifica problemele potențiale în timp util, de a le rezolva într-un termen rezonabil și de a preveni persistarea sau dezvoltarea problemelor.

5.1.4 Testarea de funcționalitate

Acest tip de testare a fost utilizat pe parcursul implementării proiectului. După fiecare funcționalitate nou creată sau pagină nou adăugată, am realizat testarea. În plus, inclusiv după modificarea bazei de date am verificat ca toate atributele să apară în interfață.

5.2 Cazuri de testare

Cum s-a menționat și anterior, s-au realizat diferite cazuri de testare pentru a performa aplicația la cote maxim. Pentru început am realizat câteva Test Case-uri. Acestea se definesc ca un conținut de informații despre acțiunile care trebuie efectuate, valorile de intrare, așteptările rezultatelor și pașii necesari pentru a valida funcționalitatea sau pentru a identifica problemele.

DIGITALIZAREA FLUXULUI DE LUCRU DINTR-O CLINICĂ DE ONCOLOGIE

Test Case ID	CM_001	Test Case Description	Testarea butonului de autentificare			
Created By	Tătar Gabriela-Diana	Reviewed By	Tătar Gabriela-Diana	Version	1	
Tester's Name	Tătar Gabriela-Diana	Date Tested	11.04.2023	Test Case (Pass/Fail/Not)	Pass	
S #	Prerequisites:		S #	Test Data		
1	Baza de date pentru aplicație		1	Nume utilizator = andrey		
2	Visual studio code		2	Parola = 123456		
3			3			
4			4			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Deschiderea server-ului Xampp, și a porturilor necesare pornirii aplicației.	Aplicația se va deschide și va afișa pagina de Acasă.	Aplicația a fost deschisă	Pass		
2	Se apasă butonul de "autentificare", fără a se introduce nicio credențială.	Ar trebui să rămână în pagina de autentificare și să afișeze mesaj de eroare.	Rămâne în pagina de autentificare și afișează mesajul de eroare "este obligatoriu", pentru ambele câmpuri care trebuiau completate.	Pass		
3	Se introduc datele pentru autentificare conform cerințelor.	Credențialele au fost introduse.	Credențialele au fost introduse cu succes, urmând intrarea în platforma web.	Pass		

Tabel 3. Test Case 1 - Testarea butonului de autentificare

Test Case ID	CM_001	Test Case Description	Testarea butonului de programare			
Created By	Tătar Gabriela	Reviewed By	Tătar Gabriela-Diana	Version	1	
Tester's Name	Tătar Gabriela	Date Tested	03.05.2023	Test Case (Pass/Fail/Not)	Pass	
S #	Prerequisites:		S #	Test Data		
1	Baza de date pentru aplicație		1	Nume utilizator = andrey		
2	Visual studio code		2	Parola = 123456		
3			3			
4			4			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Selecția unui cadru medical dorit.	Aplicația ar trebui să afișeze cadrul medical dorit.	Aplicația afișează cadrul medical dorit.	Pass		
2	Apăsare buton "programare".	Ar trebui să se redirecționeze spre pagina de "programare".	Aplicația se redirecționează către pagina de programare.	Pass		
3	Se introduc datele pentru completarea unei programări.	Datele au fost introduse.	Datele au fost introduse cu succes.	Pass		
4	Apăsare buton "programare".	Ar trebui să se redirecționeze spre pagina de unde ți-ai ales	Aplicația se redirecționează către pagina de unde ți-ai ales cadrul medical.	Pass		

Tabel 4. Test Case 2 - Testarea butonului de programare

Test Case ID	CM_001	Test Case Description	Testarea butonului de autentificare					
Created By	Tătar Gabriela	Reviewed By	Tătar Gabriela-Diana	Version	1			
Tester's Name	Tătar Gabriela	Date Tested	11.04.2023	Test Case (Pass/Fail/Not	Pass			
S #	Prerequisites:		S #	Test Data				
1	Baza de date pentru aplicație		1	Nume utilizator = andrey				
2	Visual studio code		2	Parola = 123456				
3			3					
4			4					
Step #	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended			
1	Apăsare buton "delogare".	Aplicația va ieși din site-ul web.	Utilizatorul a ieșit din contu său.		Pass			
2	Afișare pagină Acasă din perspectiva vizitatorului.	S-a redirecționat către pagina Acasă.	Utilizatorul devine vizitator.		Pass			

Tabel 5. Test Case 3 - Testarea butonului de programare

În cele din urmă, am realizat câteva cazuri de testare pe verificarea conexiunilor rutelor realizate în Postman. Am ales să facem doar patru cazuri, fiind pentru un câte un HTTP diferit. Conform capitolului 4.2.2 Pregătirea zonei de back-end, vom face pentru rutele respective.

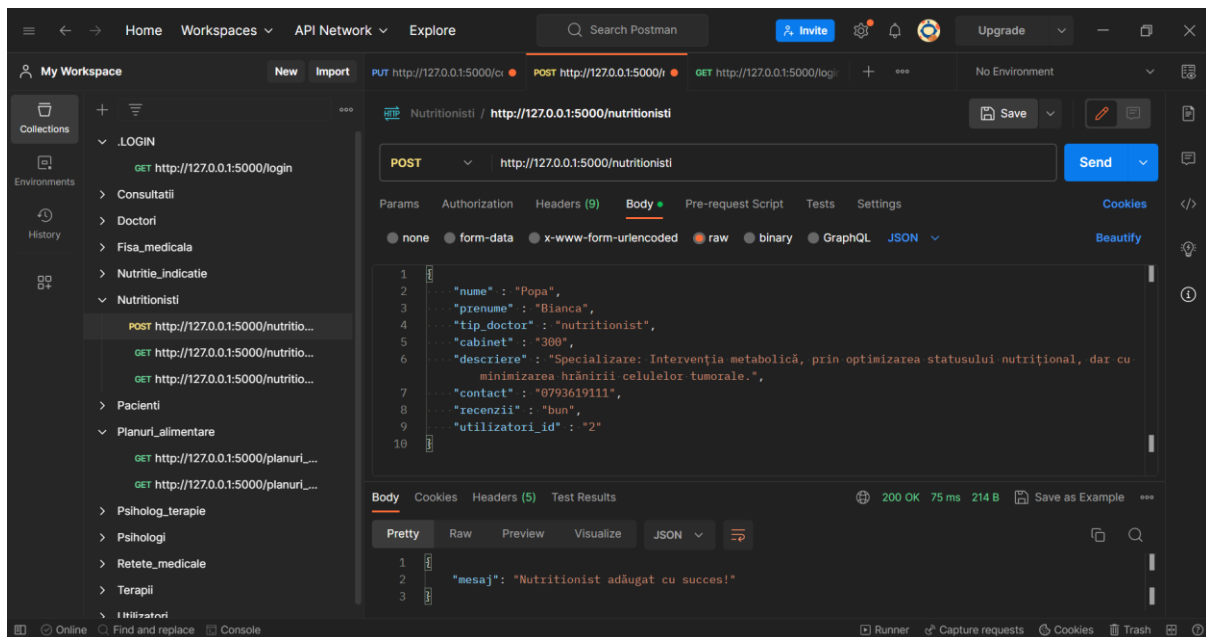


Figura 38. Testare rută POST

DIGITALIZAREA FLUXULUI DE LUCRU DINTR-O CLINICĂ DE ONCOLOGIE

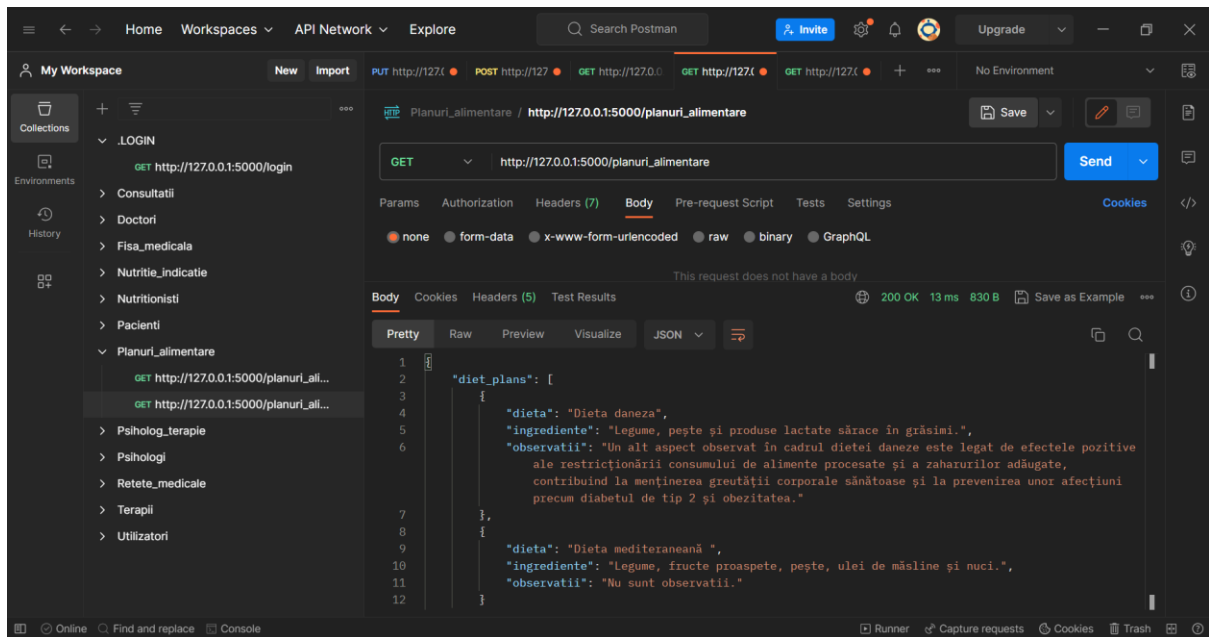


Figura 39. Testare rută GET

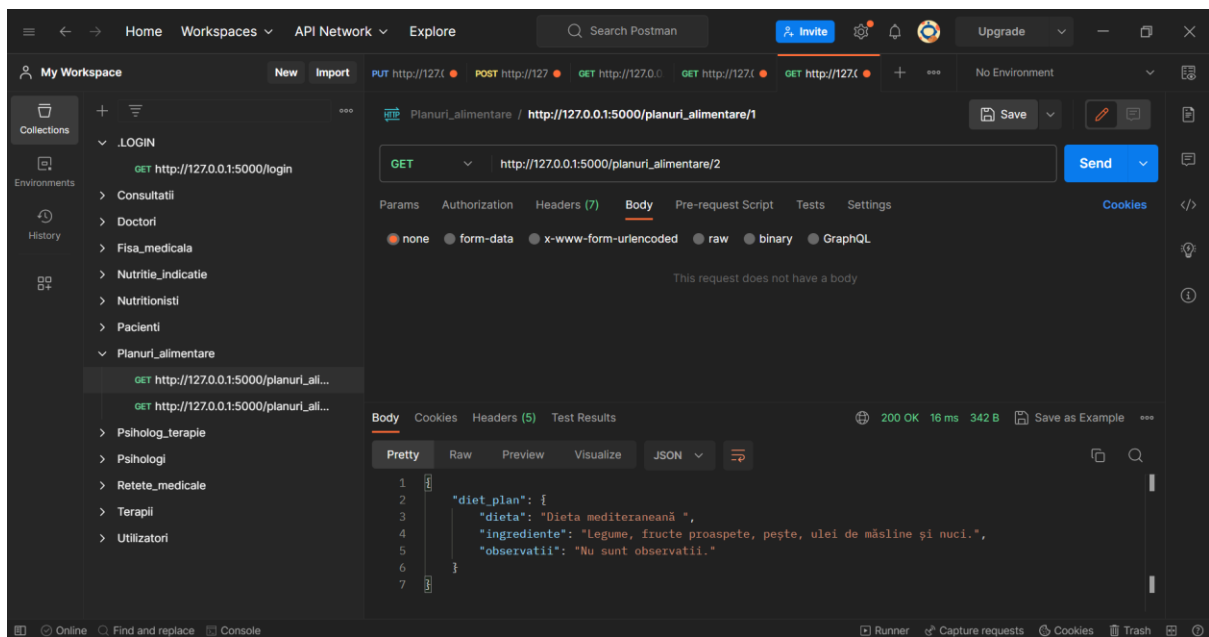


Figura 40. Testare rută GET/id

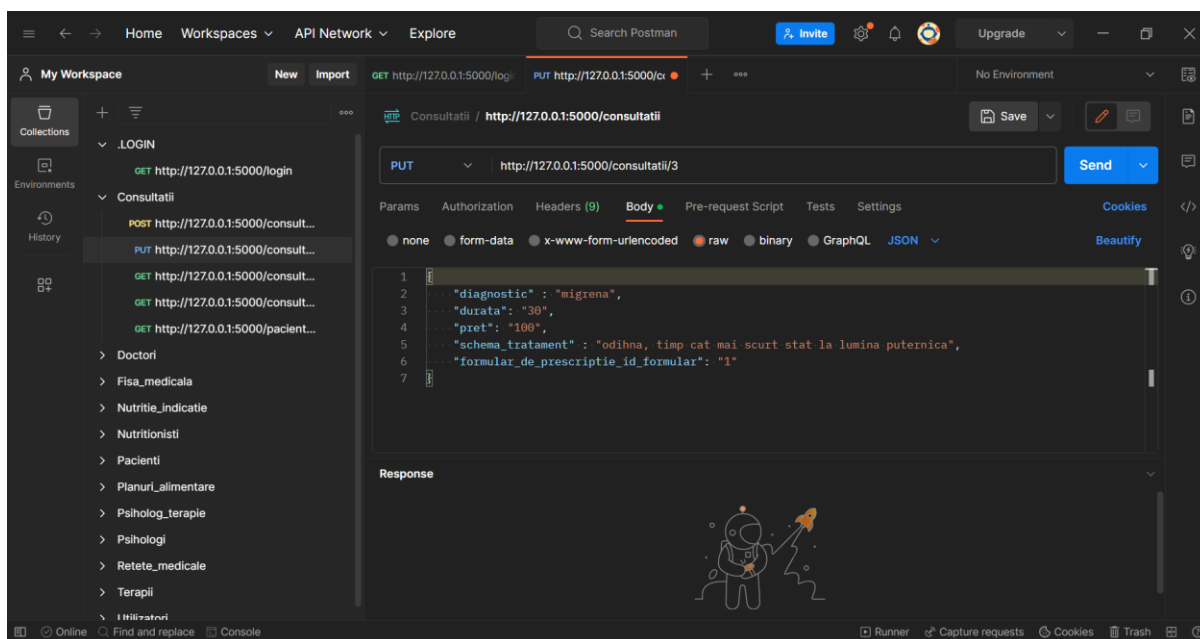


Figura 41. Testare rută PUT

Concluzii

Așadar, pe baza detaliilor prezentate pe parcursul documentației putem trage mai multe concluzii. În primul rând, există o mare nevoie de o îmbunătățire a serviciilor medicale în sectorul oncologic din România. Puținele clinici oncologice disponibile nu oferă toate serviciile necesare pacienților oncologici, cum ar fi consiliere psihologică și servicii de nutriție adecvate, ci doar pe cele medicale. În plus, deoarece prescripțiile sunt furnizate în scris, pot exista inconveniente și dificultăți în gestionarea informațiilor.

Pentru a aborda aceste probleme, s-a propus dezvoltarea unei aplicații web pentru clinicile de oncologie care să faciliteze comunicarea dintre pacienți și personalul medical. Aplicația permite programarea online a programelor și oferă informații despre medici și specialitățile acestora, precum și disponibilitatea nutriționiștilor și psihologilor specializați în oncologie. Digitalizarea fișei medicale a pacientului va ajuta la o mai bună organizare a echipei medicale.

Obiectivele acestei aplicații includ crearea unui mediu confortabil și sigur pentru pacienții aflați în tratament, dezvoltarea programelor continue pentru cei diagnosticați, furnizarea de servicii nutriționale adecvate și consiliere psihologică, creșterea numărului de personal medical, include îmbunătățirea satisfacției pacienților și a personalului medical. De asemenea, se evidențiază importanța protejării mediului înconjurător, prin gestionarea deșeurilor și utilizarea materialelor prietenoase mediului.

Aplicațiile de succes necesită dezvoltare continuă, îmbunătățiri ale interfeței cu utilizatorul și testări permanente. Pentru a accesa aplicația sunt necesare o conexiune la internet și un browser. Datele medicale sunt stocate într-un mediu securizat pentru a preveni pierderea sau furtul datelor.

Din viziunea tehnologiile folosite pentru implementarea unei aplicații web pentru o clinică de oncologie medicală a fost benefică din mai multe motive. Utilizarea limbajului de programare Python împreună cu framework-ul Flask a permis gestionarea eficientă a logicii și funcționalității aplicației, precum și procesarea datelor din bazele de date. Flask oferă un mediu ușor de utilizat și extensibil, care permite dezvoltatorilor să aleagă instrumentele și bibliotecile potrivite pentru proiectele lor. Angular a fost, de asemenea, o alegere bună pentru front-end, deoarece oferă un cadru bazat pe componente și o gamă largă de biblioteci pentru construirea de interfețe web robuste și eficiente. Utilizarea Visual Studio Code ca mediu de programare a

oferit suport puternic și scalabil pentru dezvoltarea aplicațiilor, ajutând dezvoltatorii să fie mai eficienți și mai productivi. Bootstrap permite dezvoltatorilor să creeze rapid și eficient grafică uniformă și consecventă în aplicațiile lor, indiferent de dispozitivul utilizat. Prin urmare, aceste alegeri tehnologice au oferit multe avantaje pentru implementarea cu succes a sistemelor informatice în clinicile de oncologie.

În ultima instanță, dezvoltarea unei aplicații web pentru o clinică de oncologie poate ajuta la îmbunătățirea serviciilor medicale, la creșterea satisfacției pacienților și a personalului medical și la crearea unor relații pozitive între aceștia.

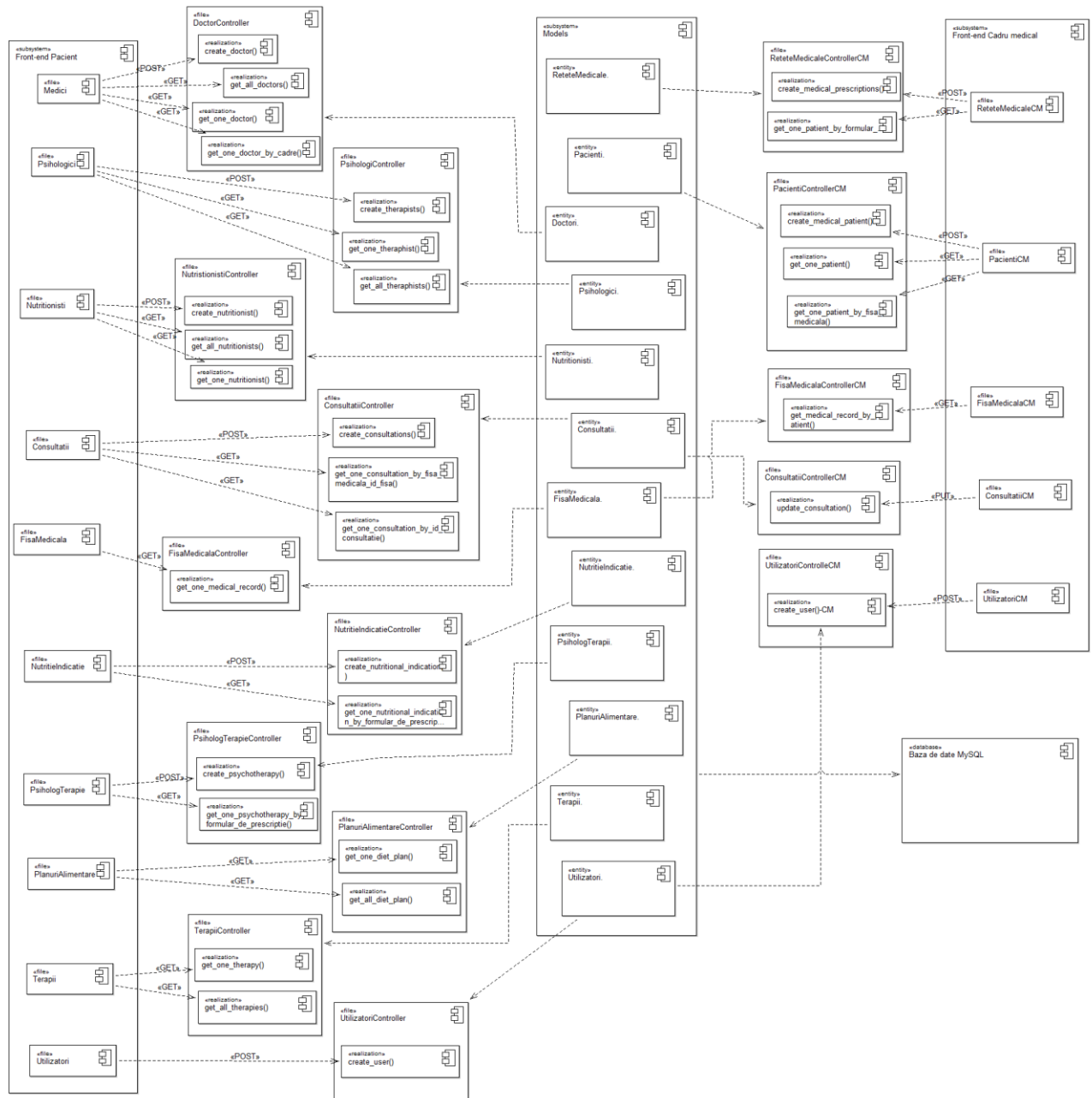
Dezvoltarea aplicației în viitor este un pas pe care îl luăm în considerare, deoarece orice produs poate fi îmbunătățit. O idee de punere în practică, cu trecerea timpului, ar fi implementarea unui chat, unde pacienții ar putea pune diverse întrebări pentru a le satisface curiozitatea în legătură cu domeniul medical oncologic. Mai mult, un obiectiv mare ar fi acela de a realiza aplicația și pe mobile.

În momentul actual aplicația are doar doi actori, unul de vizitator și altul de utilizator (cadrul medical și pacient), putând fi adăugat și un administrator al site-ului. Acesta ar avea diferite sarcini, cum ar fi: întreținerea platformei unde să poată adăuga, modifica sau șterge lucrurile dorite, optimizarea aplicației. Ar fi indicat să se dezvolte și o parte de administrator, pentru a se apela pe urmă la dezvoltatorul aplicației doar în momentele în care apare o problemă complexă.

Din punct de vedere al codului, partea de back-end ar putea fi gestionată și împărțită mai structurat, pentru că în momentul actual sunt doar două fișiere (`app.py` și `models.py`). În plus, parolele utilizatorilor ar putea să fie criptate în baza de date. Acesta este o viziune bună, deoarece o parolă criptată asigură securitatea și protejează informațiile sensibile împotriva accesului neautorizat și utilizării frauduloase.

Anexe

Anexa 1. Diagrama de componente completă



Bibliografie

- 3rohealthreview. (2018, noiembrie 15). *Impactul Digitalizării ASUPRA serviciilor medicale*, <https://rohealthreview.ro/impactul-digitalizarii-asupra-serviciilor-medicale/>
- Angular. (2010-2023). *What is Angular?*, <https://angular.io/guide/what-is-angular>.
- Anna, O. (2019, octombrie 29). *Public health digitalization in Europe*, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6859512/>
- Balooni, A. (2021, iunie 9). *Zenesis*, <https://www.zenesis.com/everything-you-should-know-about-bootstrap-5>
- Buchmann, R., & Ghiran, A.-M. (2023). *Carte suport licență Web Semantic*.
- Clauser, S. B. (2011). Improving Modern Cancer Care Through Information Technology. *American Journal of Preventive Medicine*, pg. 198-207.
- Gourley, D., Totty, B., Sayer, M., Aggarwal, A., & Reddy, S. (2002). *HTTP: The Definitive Guide*. O'Reilly Media, <https://www.codecademy.com/article/what-is-an-ide>.
- Jecan, S. (2022). *Curs opțional Testarea Produselor Soft*.
- Koffer, P. (2022, ianuarie 20). *What is the Angular framework?*, <https://mdevelopers.com/blog/what-is-the-angular-framework>
- Kreps, G. L. (2010, ianuarie 21). New directions in eHealth communication: Opportunities and challenges. *Patient Education and Counseling*, pg. 329-336.
- Microsoft. (2023). *Getting Started*, <https://code.visualstudio.com/docs>
- Oracle. (2023). *Ce Este O Bază de date Relațională?*, <https://www.oracle.com/ro/database/what-is-a-relational-database/>
- Pallets. (2010). *Installation*, <https://flask.palletsprojects.com/en/2.3.x/installation/>
- Pallets. (fără an). *Flask*, <https://palletsprojects.com/p/flask/>
- phpmyadmin. (2003-2023). *Bringing MySQL to the web*, <https://www.phpmyadmin.net/>

Science, I. (2023, iunie). *Ce Este arhitectura PE trei Niveluri? - definiție din techopedia*,
<https://ro.theastrologypage.com/three-tier-architecture>

Team, C. (2023). *What Is an IDE?*, <https://www.codecademy.com/article/what-is-an-ide>

Wikipedia. (2023, iunie 14). *Angular (web framework)*,
[https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework))